*(continued from part 39)*

### Timing and control

The clock and instruction execution times must be fast enough to perform the required task in the time available. This consideration is also important for the other functional devices.

If the microprocessor control signals are such that they may be connected *directly* to the memory and input/output devices, system timing is made easier because time delays through additional circuits are avoided.

By ensuring timing and control compatibility within a microprocessor based system, further advantages are obtained, however, because system design time and cost may be reduced, and the cost of additional circuits is also saved.

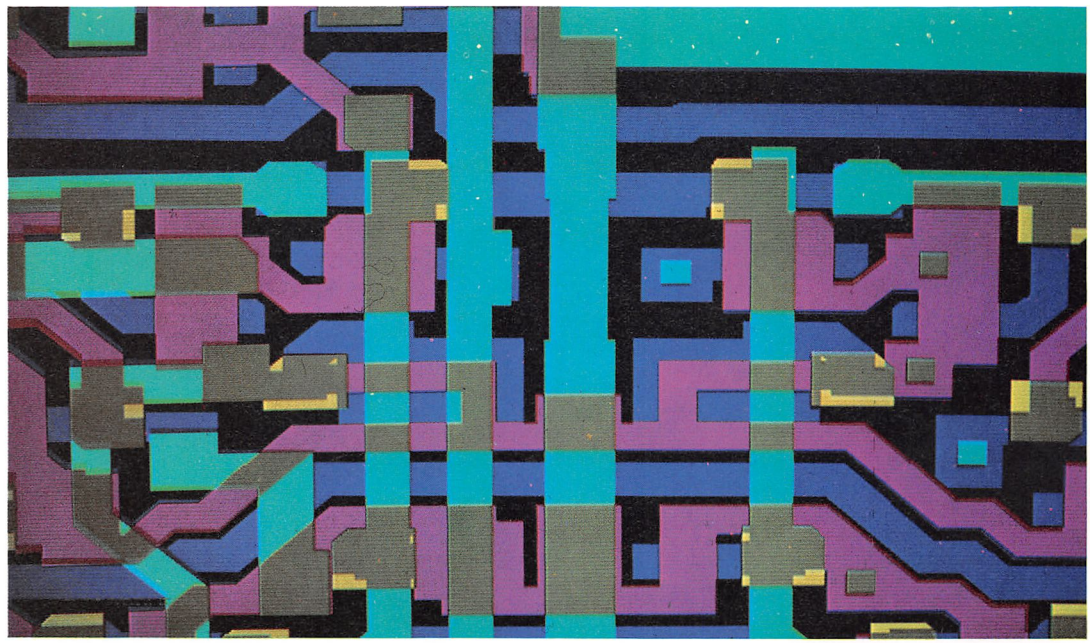
### Interrupt sequence

The microprocessor within a system generally performs one program instruction after another, continuing until terminated with a specific STOP or HALT instruction. However, most microprocessors have an instruction in their instruction sets which allows the program to be interrupted at unexpected, random times. Such an **interrupt** instruction could, for example, be used if it is important for a particular input device to communicate with the microprocessor.

Many microprocessors do not accept interrupt signals in a totally random way, however. Generally, upon receipt of an interrupt, the microprocessor continues with the program in hand until it receives an instruction that tells it to check for an interrupt signal. In effect, the microprocessor is only interrupted if the program allows it to be. Some more advanced microprocessors have an **interrupt sequence** which does respond to random interrupts in a more random way.

How fast a microprocessor responds to an interrupt determines whether it is acceptable or not for many situations. If input information must be received often at unpredictable times, then the programmed task the microprocessor is performing is correspondingly often interrupted.

In the example of the noughts and crosses game, the only time the microprocessor needs to be interrupted is when the person makes a move, or when a new game is started. The speed with which the microprocessor system's opponent makes a move is so slow in comparison to the microprocessor itself, that the interrupt procedure is, in fact, unimportant in this case. On the other hand, if a microprocessor receives information from, say, a satellite communications system and therefore has to respond very quickly, the interrupt procedure is of critical importance.



**Right:** computer graphic, showing the layout of circuitry within a Z8000 microprocessor. (Photo: SGS).

## Microprocessor versus microcomputer

To provide the sense, decide, remember and act functions of a computer system, a microprocessor (the decide function) must have input, output and memory functional devices around it, forming a complete computer system.

ICs which have all of these functions on a *single* chip, on the other hand, are known as **microcomputers**. (The name microcomputer is often, confusingly, also used to describe a computer system built using a microprocessor – but strictly speaking a microcomputer is an IC.)

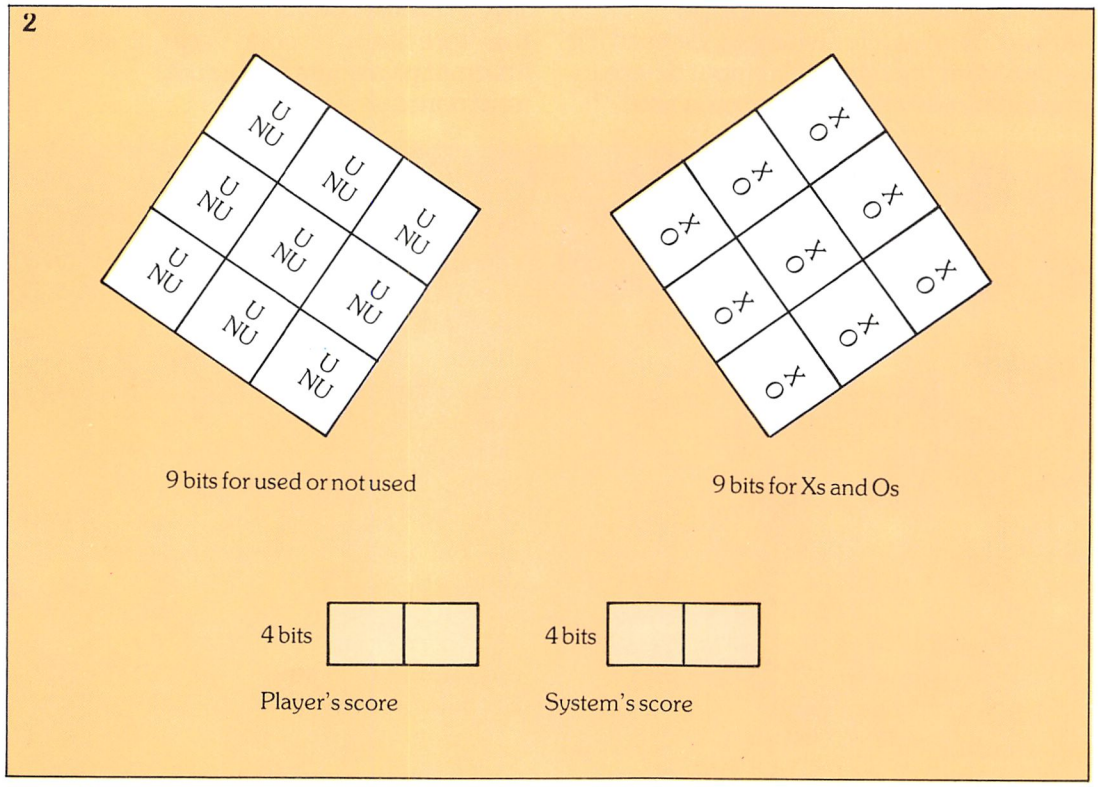
As long as a microcomputer is able to satisfy the needs of the system regarding bit-length, speed, instruction set, timing, control and interrupt, it should be considered alongside available microprocessors. Often, the system may thus be constructed using only one IC, instead of many – a very important factor on the production line. Of course, the microcomputer must also have sufficient memory and input/output capability to meet the system needs. So, when comparing a microcomputer with the equivalent microprocessor, we need to know these additional system requirements.

### Memory

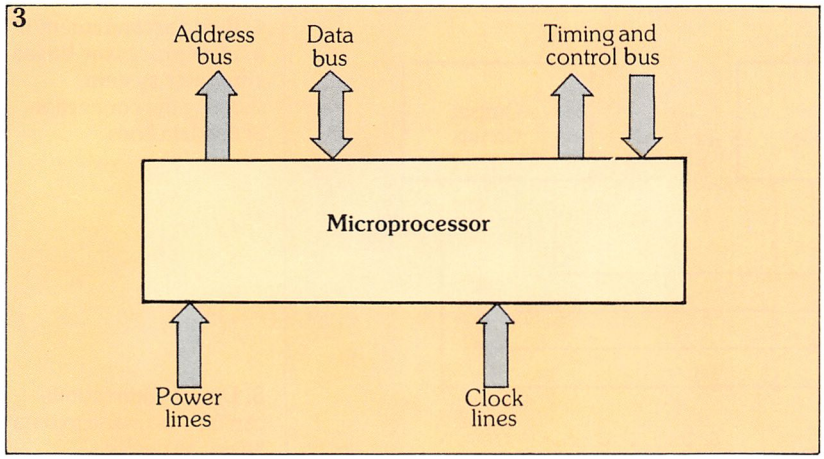
The system program is stored in memory, along with data to be used. Without defining requirements down to the last memory cell, estimates of total memory requirements must be made. Total memory requirements will, of course, consist of some ROM (to store the program) and some RAM (to store data).

Usually, the data memory requirements of the system are fairly well determined by the description of the system. For example, the noughts and crosses game should only need 18 bits of information to keep track of the Xs, 0s and squares that haven't been filled: 9 bits are used to indicate if a position is used or unused – say, 0 for used and 1 for unused; and another 9 bits are used to indicate the state of play – say, 0 for 0 and 1 for X. The player's score could be held by 4 bits – up to 16 games – and similarly the system's score by another 4 bits as shown in *figure 2*. Thus, 26 bits of RAM are required.

If a 4-bit microcomputer with at least seven internal 4-bit RAM locations is




**2. Estimating the amount of memory space** needed to store the information required in the noughts and crosses game.




**3. Most microprocessors have these signal lines,** arranged as busses.

available, there is no need for any external RAM.

The ROM requirement of the system, however, is more difficult to define, as it depends on the program length – and the program has not yet been written. However, an estimate should be attempted.

### Input/output devices

A careful review of system behaviour usually reveals what types of input and output devices are required. In our noughts and crosses game, for example, indicating lights can be used to show whether a square has a nought or a cross in it (9 for 0, 9 for X). These are system output devices.

Input devices required are an on/off switch, a switch to start a new game, and a set of switches (perhaps in a calculator-type keyboard) to allow the player to input the next move.

All of these system inputs and outputs must be interfaced to the microprocessor or microcomputer so that player and system can communicate.

More complex systems usually require more complex input/output devices and interfaces. For example, one of the principle ways of interfacing computer systems and human players is to use a keyboard and VDU. Most microprocessors, in fact, form the heads of complete families of devices – the other members of which allow interface to such things as keyboards, VDUs, auxiliary memory etc.

Once the microprocessor or microcomputer is selected for the system, and the input/output devices and requirements are known, the system may be built by interconnecting all of the individual parts.

### Connecting system components together

All microprocessors have a number of signal lines or busses which must be connected to external devices correctly if the system is to work as required. Most microprocessors have similar categories of signal lines. These are:

1) **Address lines**, forming the address bus – the digital code that appears on these lines defines the location of an instruction or data to be used next by the microprocessor.

2) **Data lines**, forming the data bus – containing the actual instruction and data codes, to and from the microprocessor.

3) **Timing and control lines**, forming the control bus – signals to and from the microprocessor, allowing it to control external functional devices. Interrupt signals are included in this category.

4) **Clock lines** – for many microprocessors the clock signals are formed externally and sent via these lines. Other microprocessors and many microcomputers have internal circuits which generate these signals.

5) **Power lines** – the electrical current and operating voltages required by the microprocessor are carried by these lines.

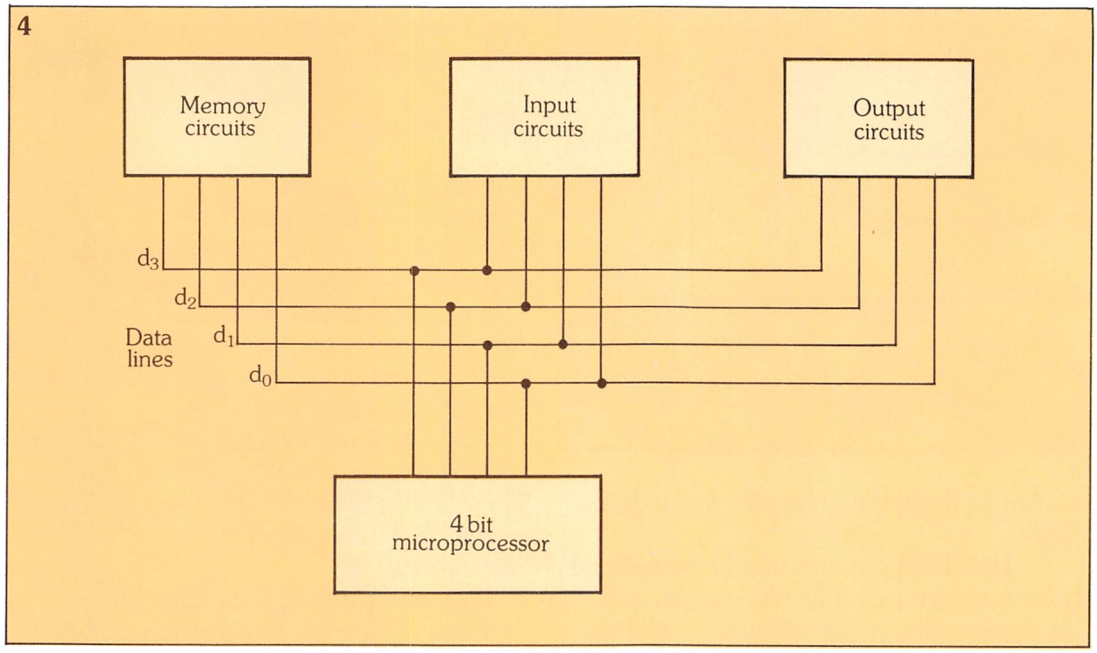
*Figure 3* shows these signal lines in relation to a microprocessor.

### Data line connections

A microprocessor based computer system is shown in *figure 4*. As we can see, the data lines connect many circuits to the microprocessor. However, all connected circuits have their data lines connected in the order shown – all the least significant bit lines ($d_0$) are connected together; all of the next significant bit lines ($d_1$) are connected together, and so on. This is true whether the microprocessor is a 4-bit device as in *figure 4*, or an 8-bit, 16-bit or 32-bit microprocessor.

Such connections may be made directly only if all devices have **three-state outputs**, as shown in detail in *figure 5*.

*Figure 5a* shows the output of a latch which is to feed bits of information onto




**4. Basic arrangement of a microprocessor based computer system,** showing the connection of the data lines.

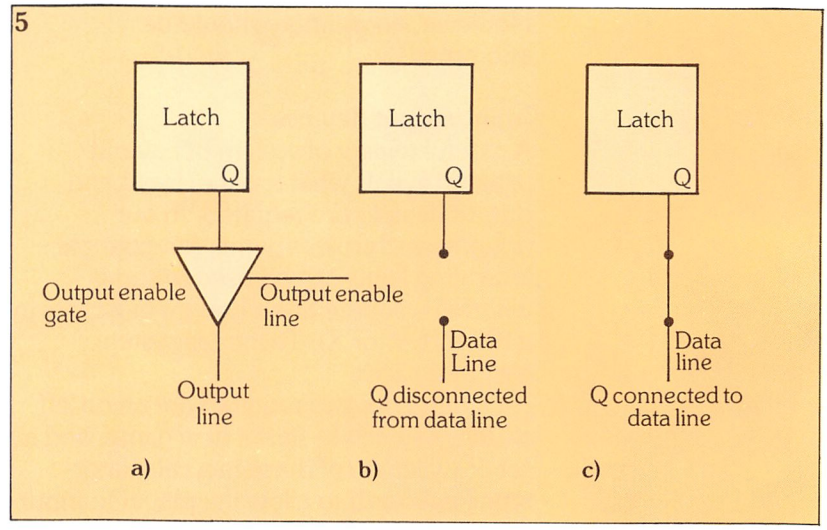
**5. Direct connections** can only be made to data lines if all devices involved have three state outputs: **(a)** three state output; **(b)** effective circuit with output disconnected; **(c)** effective circuit with output connected.

one of the data lines. An **output enable gate** is positioned between the latch and the data line. If the logic state of the **output enable line** is 1, the output enable gate is inactive, and so the latch output is effectively disconnected from the data line, as shown in *figure 5b*. The data line is therefore not affected by the logic state of the latch output – whether it is a 0 or a 1.

If the logic state on the output enable line is 1, however, the output enable gate is active, and the gate acts as a short circuit between the latch output and the data line. So, if the latch output is 1, the logic state of the data line is 1, if the latch output is 0, so is the data line.

These, then, are the three states of a three-state output – open-circuit, 1 and 0. But what use are they? We can find this out by considering in detail the output stage of a non three-state device and comparing it with a typical device used to form a three-state output.

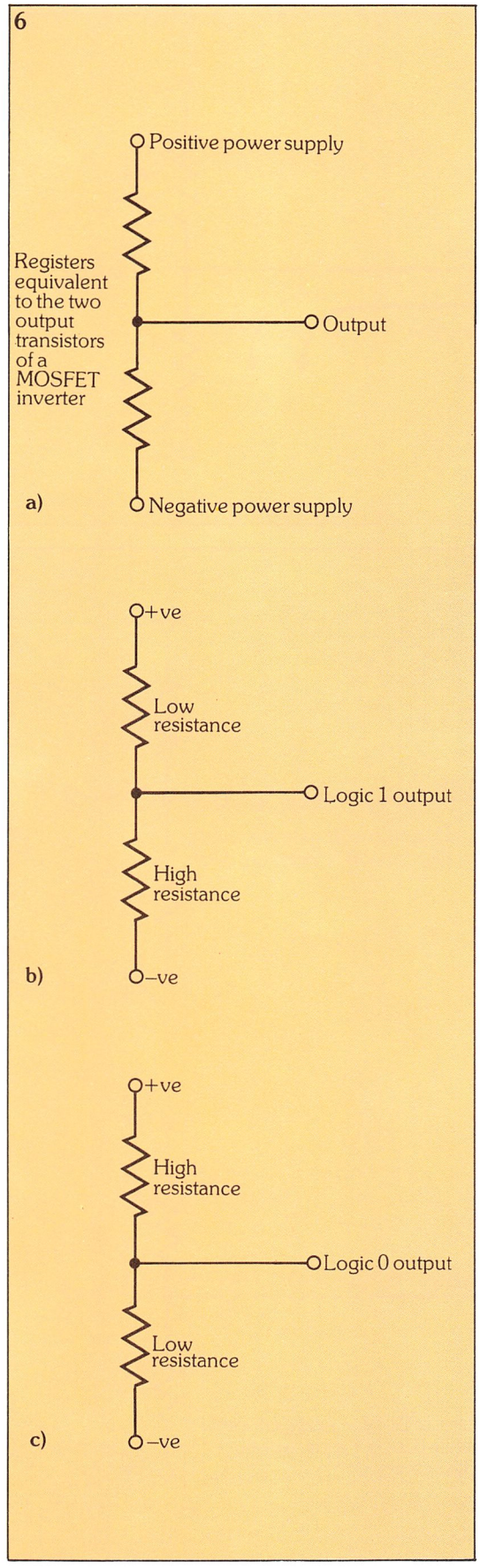
In *figure 6*, the equivalent circuits of an output stage of a MOSFET inverter are shown. Now, an ordinary MOSFET inverter has an output stage which may be considered to consist of two transistors in series. The output connection is where the two transistors connect. As the two transistors are either off or on, we may consider them to be represented by two equivalent resistors (as in *figure 6a*) which have a high value or a low value




depending on whether the represented transistor is off or on.

If, say, the data line of a microprocessor system is connected to the output of the inverter, as shown in *figure 6b*, and the inverter's bottom transistor is off while the top transistor is on, then the bottom resistance is high, while the top resistance is low. Therefore the voltage on the data line is almost that of the positive supply, i.e. logic 1.

When the opposite occurs, and the top transistor is off while the bottom transistor is on, the output logic state is 0. This is because the top resistance is of high value while the bottom resistance is of low value (*figure 6c*).




**6. (a) Equivalent circuit** of a MOSFET inverter output stage; **(b)** equivalent circuit for an output of logic 1; **(c)** equivalent circuit for an output of logic 0.

So far so good. But what happens if more than one such device is connected to the same data line? *Figure 7* illustrates the situation when two output stages are connected to a single data line. One of the output stages has its top resistance of low value, and its bottom resistance of high value – thus it is trying to cause a logic state of 1. The other output stage, however, has its top resistance of high value and its bottom resistance of low value – it tries to cause a logic state of 0 on the data line *at the same time.*

The overall effect of this is that a large current flows from device 1 to device 2 along the data line, the voltage on the data line is unknown – it could be anywhere between the positive and negative supply voltages – and the output stages of both devices may be damaged.

*Figure 8* shows that three-state output devices may overcome these problems, allowing any number of devices to be connected to the same line, simultaneously. *Figure 8a* shows how device 1 is electrically connected to the data line, while device 2 is electrically disconnected (i.e. device 2 is in the open-circuit output state). Similarly, *figure 8* illustrates how device 2 is electrically connected while device 1 is electrically disconnected. Each device is electrically connected by enabling its output, using the output enable gate. Any number of output devices may be connected to the data line *as long as no more than one output is enabled at any one time.*

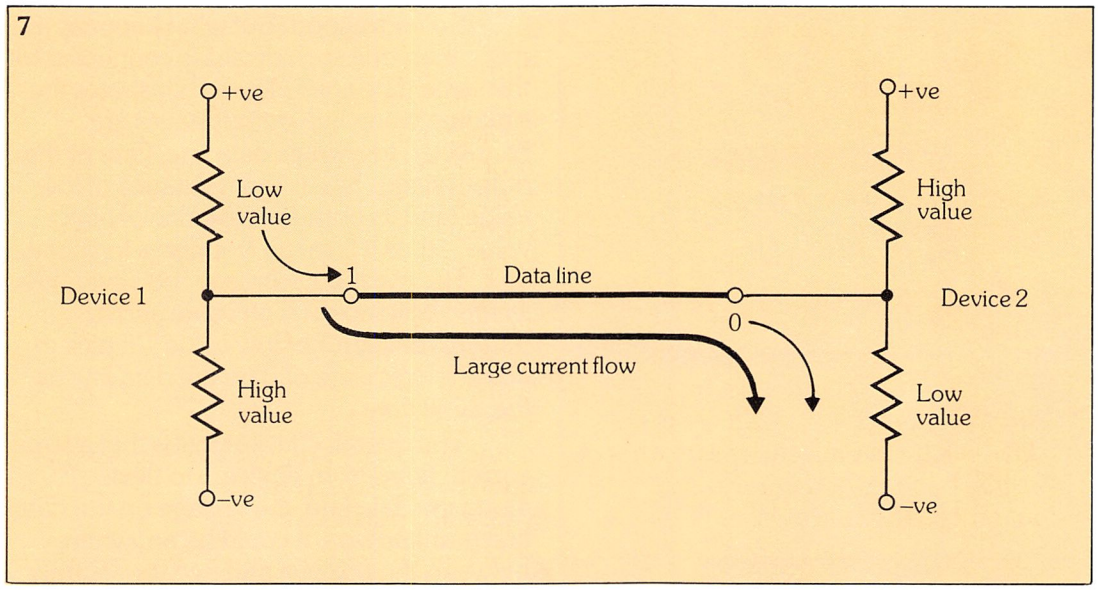
### Address line connections

The connections of the address lines in most microprocessor systems are achieved very simply, as shown in *figure 9*. Address signals generated by the microprocessor are sent to the address input pins of the other functional devices, along the address bus.

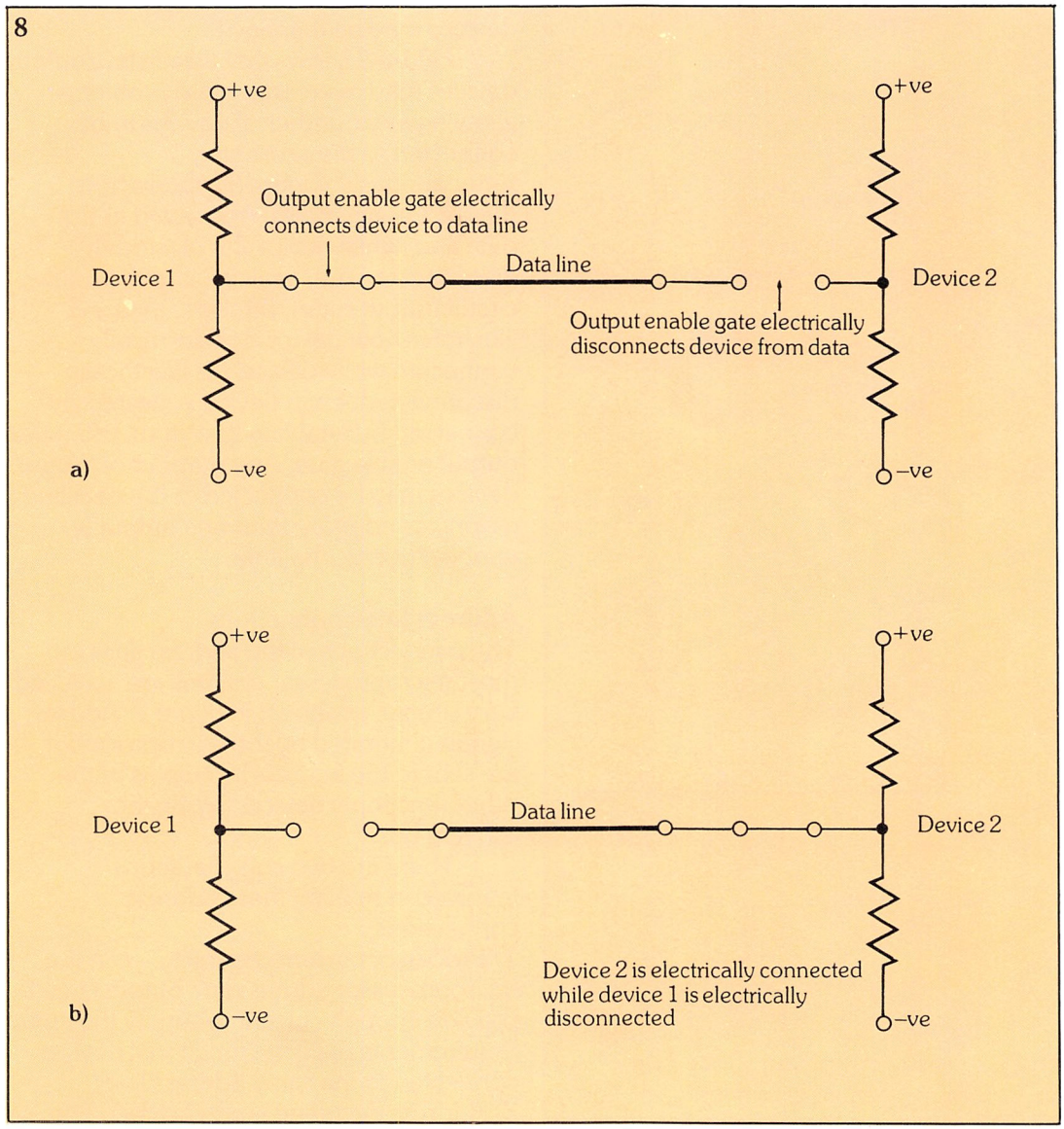
There are two considerations involved in making these address connections:

1) Processor current capability – can the microprocessor address line output circuit provide enough current to supply the input requirements of *all* the functional devices connected to the same address line?

2) Address decoding – are the decoding




**7. Equivalent circuit** showing the situation when two devices are connected to the same data line.




**8. (a) Device 1 is electrically connected** to the data line, while device 2 is disconnected; **(b)** device 1 is now disconnected, while device 2 is connected.

circuits inside the functional devices capable of decoding the address signals or do they require help from outside decoding circuits?

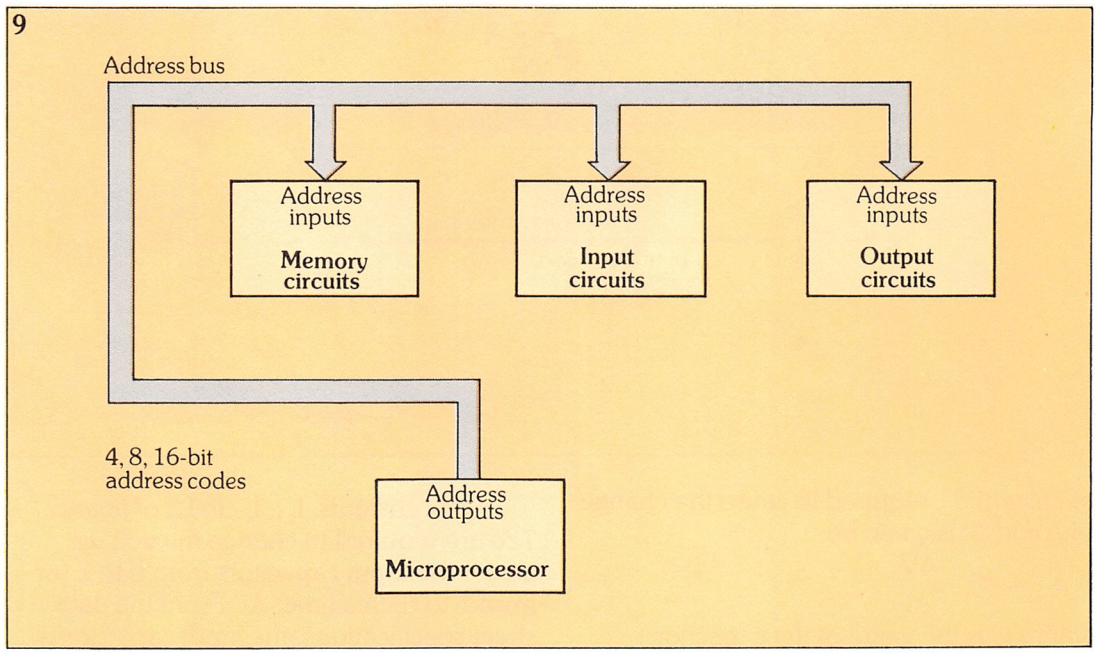
We shall look at the first consideration now, but leave the second until *Microprocessors 4*.

### Processor current capability

Microprocessor output circuits have the ability to supply current to a line or to receive current from a line. The maximum or minimum limits of this current are specified by the manufacturer's data sheet. The address inputs of memory and input/output functional devices have to be supplied a certain level of current in the 0 input case and another level of current in the 1 input case. These currents are similarly specified by the manufacturer on data sheets. In addition, supplying or receiving current depends on the level of the output.

Consider *figure 10*. When the processor output is in the 1 state, current flows from the processor to the input device and this is supplied by the processor which is said to be **sourcing** current. When the processor output is in the 0 state, on the other hand, the current flows from the input device to the processor and the processor must be able to receive this level of current without damaging its output circuit. The processor, in receiving this current, is said to be **sinking** current.

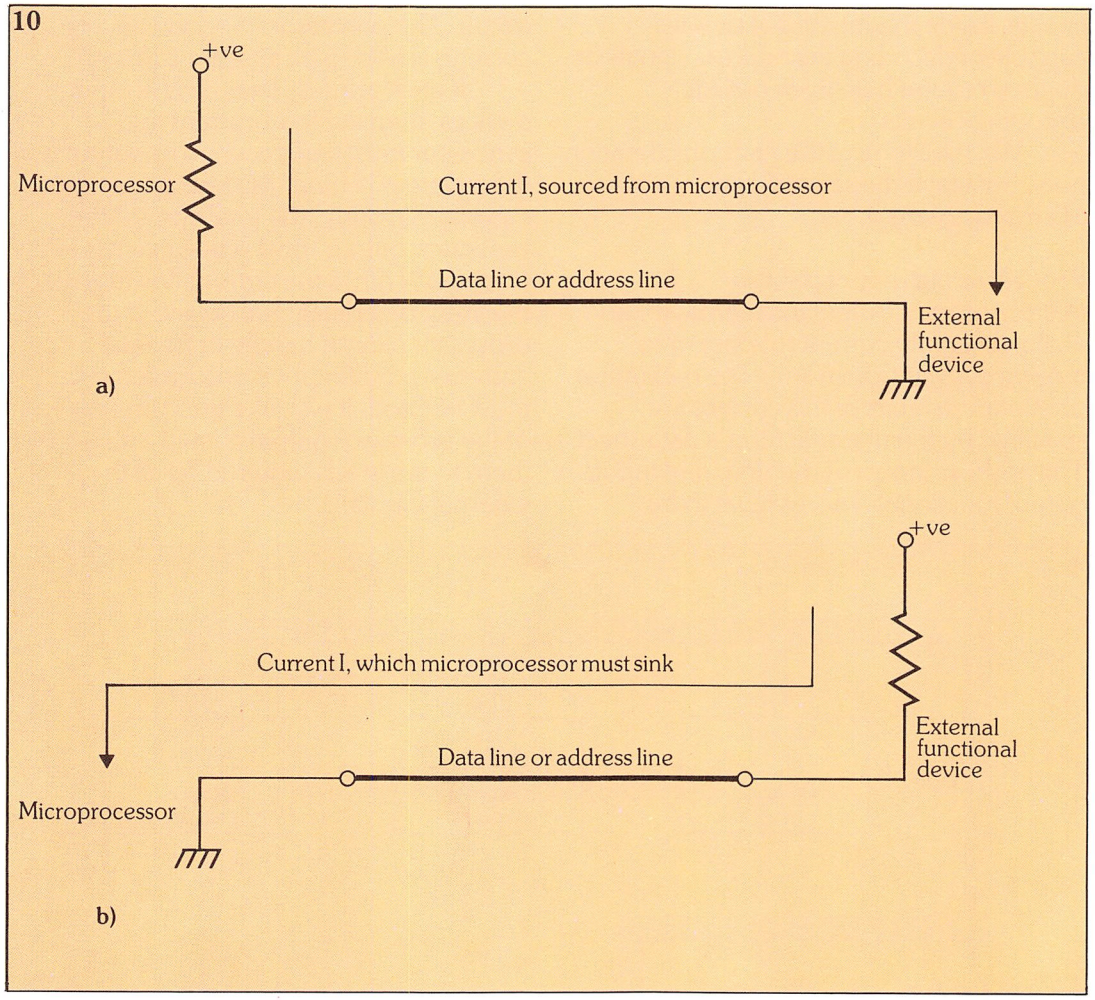
If each address line feeds N input devices, then the total current the processor must source when its output is in the 1 state is N times the individual input current. Similarly, the total current the processor output must sink when it is in the 0 state is N times the current flowing out of the input of a functional device. If the processor cannot handle these total currents, a buffer driver integrated circuit must be used. It will take the current levels of the processor output circuits and boost them to the levels required by all the external inputs.




**9. Address signals** generated by the microprocessor are sent to the address inputs of other devices, along the address bus.

There is another feature of the functional devices' inputs that affects the sink and source currents required. The circuits forming each input effectively act as a small value capacitor.

As we know, a capacitor is itself a form of storage element, the voltage across which may change only if the charge stored in the capacitor changes. But the stored charge may, in turn, change only if current flows into or out of the capacitor. *Figure 11a* shows a capacitor, representing an input, connected to a data or address line. *Figure 11b* shows that the voltage across the capacitor needs a certain amount of time, $\Delta t$, to allow the voltage to change a certain amount, $\Delta V$. The amount




**10. (a) When the microprocessor output** is at logic 1, it is said to be sourcing current; **(b)** at logic 0, the microprocessor sinks current.

of current, $I_c$, required to cause the change of $\Delta V$ in $\Delta t$ is given by:

$$I_c = C \frac{\Delta V}{\Delta t}$$

where C is the value of the capacitor.

**Output current required**

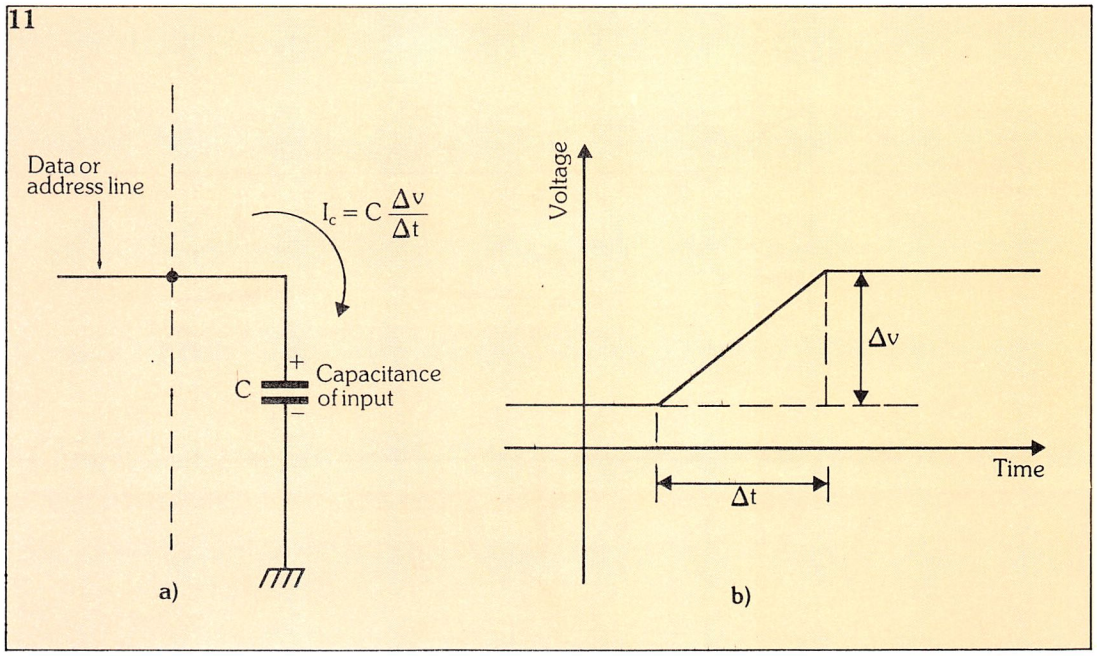
This equation can be used to determine if the processor outputs can deliver enough current. Take *figure 12a* as an example. Here, the processor's address line output connects to eight address inputs of external functional devices. Currents $I_1$, $I_2$ to $I_8$ represent the currents required by each input. From the manufacturer's data sheet for these functional devices, current $I_1$ is 0.01 mA (flowing into the input) when the address line is a 1, and it is –1 mA when the line level is an 0. The minus sign means the current is flowing in the opposite direction to the arrow in *figure 12a*. In other words, the output is sinking the current.

The currents, $I_{c1}$, $I_{c2}$ to $I_{c8}$ of *figure 12a* are required to change the voltage across the input capacitors from 0 to 1 (or from 1 to 0) in a time, $\Delta t$. From the data sheet specifications, the input capacitor is 10 pF.

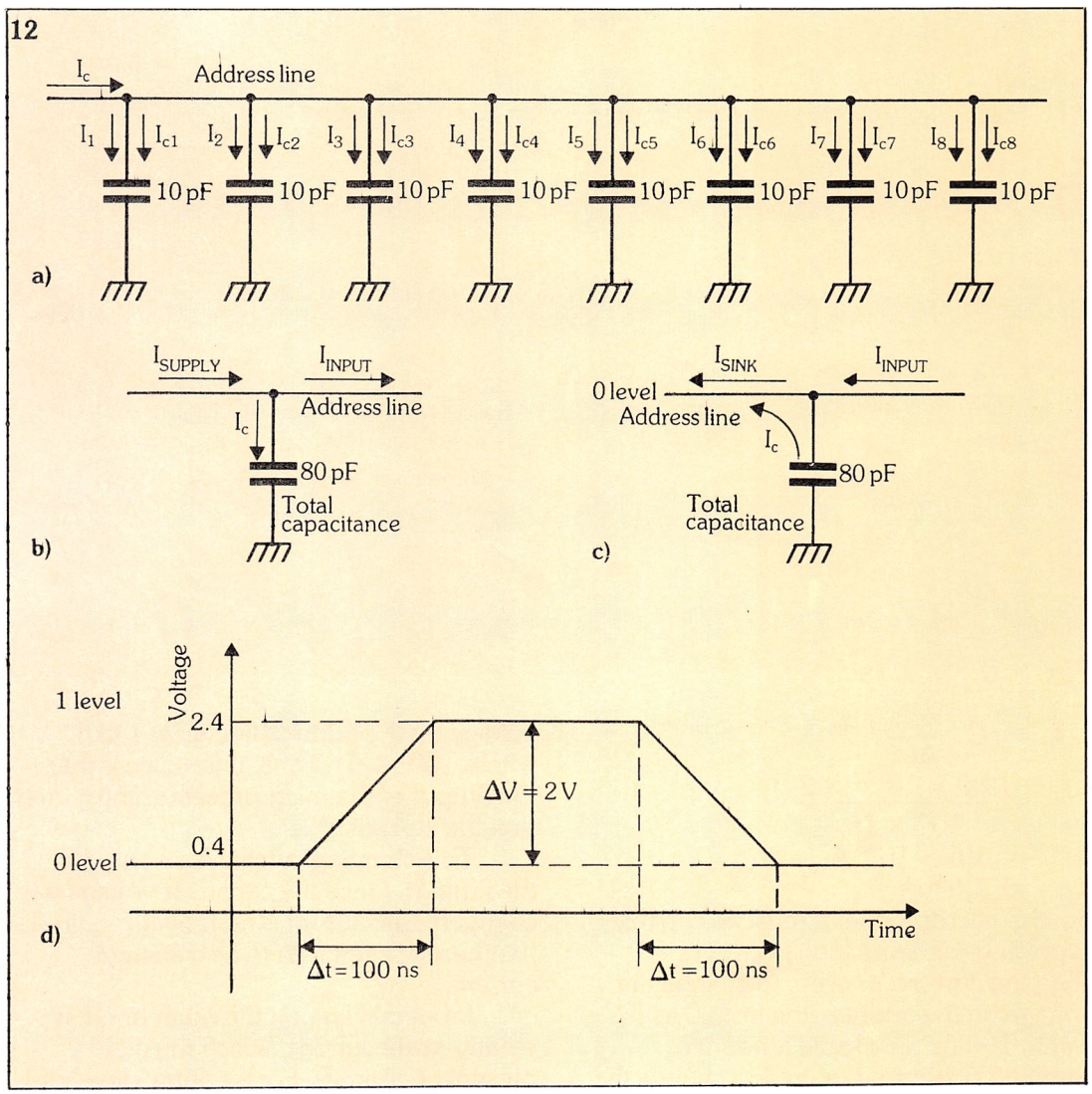
*Figures 12b* and *c* illustrate how the problem can be simplified. All of the input currents can be represented by one value, $I_{INPUT}$. All the individual capacitors can be combined into one 80 pF (8 × 10 pF) capacitor, and all charging currents can be combined into one value, $I_c$.

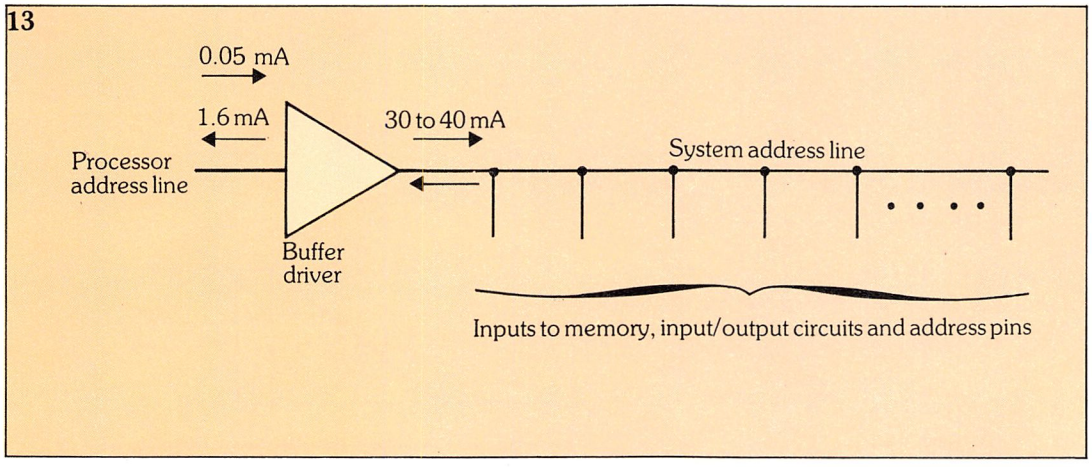
First, we'll determine what charging current the processor output must supply to change the input capacitor voltage. As shown in *figure 12d*, we'll assume that the output is at a 0 level of 0.4 V and it is to be changed to the 1 level voltage of 2.4 V (i.e. $\Delta V = 2$ V). So that input circuits will react correctly and the microprocessor can run at full speed, this voltage change must occur in, say, 100 ns. Solving the equation:

**11. (a) The circuits forming each input** act as a small capacitor; **(b)** the voltage across this capacitance needs time t, to change by an amount, V.

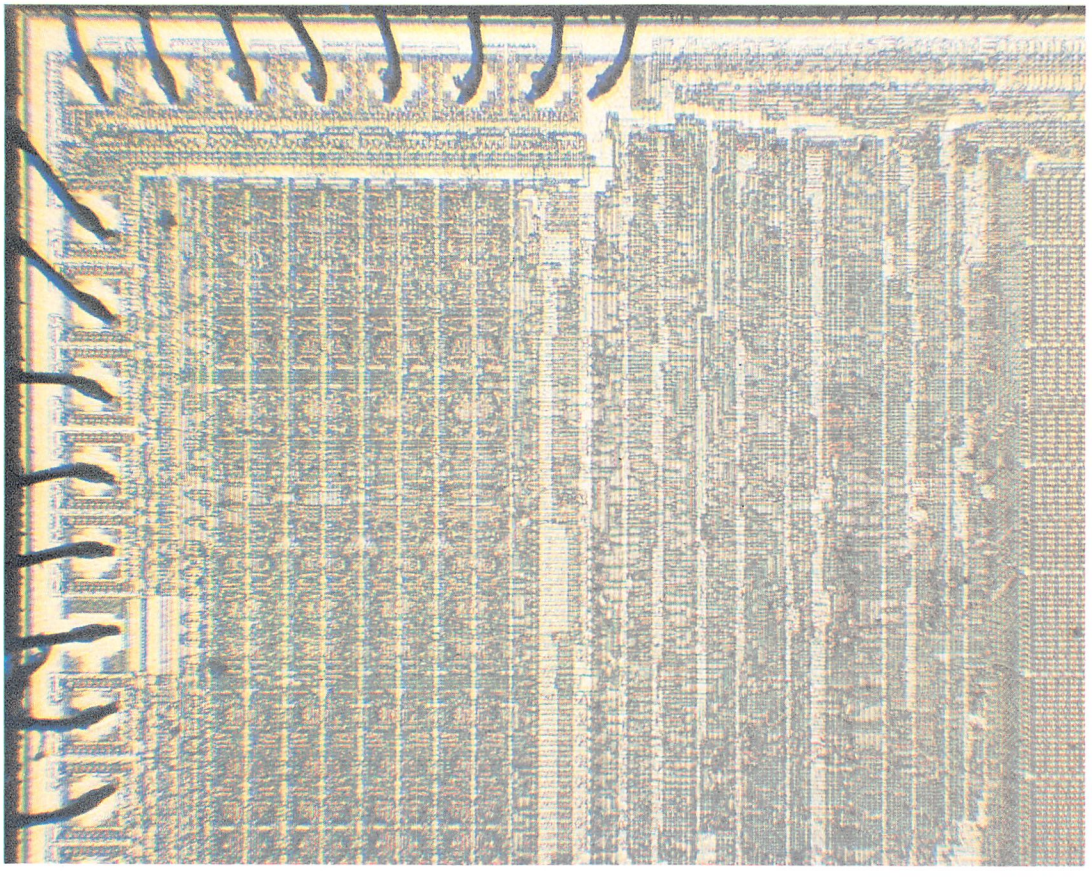



**12. (a) A microprocessor's eight address outputs**, showing capacitances and currents flowing; **(b)** representing all the output currents at logic 1, by a single value; **(c)** representing all the output currents at logic 0, by a single value; **(d)** time taken for all outputs to change logic states is 100 ns.






**13. If the microprocessor output** cannot source or sink the required currents, then buffer drivers have to be added.



**Left:** close up of a microprocessor chip.

Neville Miles

$$I_c = C \frac{\Delta V}{\Delta t}$$
$$= \frac{80 \times 10^{-12} \times 2}{100 \times 10^{-9}}$$
$$= 1.6 \times 10^{-3}\,A$$
$$= 1.6\,mA$$

So, the microprocessor must be capable of supplying (i.e. sourcing) 1.6 mA of charging current, in order to change the voltage on the address line from 0 to 1 within 100 ns. The same amount of charging current is required to change the voltage on the address line from 1 to 0 within 100 ns. The only difference is that the output of the microprocessor must then sink the current.

This charging current lasts only for the time $\Delta t$. Once the capacitor voltage has changed, the current is no longer required. It is therefore referred to as **transient** current.

Input current, on the other hand, is **steady-state** current, which must be present at all times. For an output level of 1,

Table 2
**Summary of output current requirements**

| | $I_c$ | $I_{INPUT}$ | Microprocessor current |
|---|---|---|---|
| 1 level | 1.6 mA | 0.08 mA | 1.68 mA |
| 0 level | –1.6 mA | –8 mA | –9.6 mA |

the amount of current that must be sourced is given by:

$$I_{INPUT} = 8 \times 0.01$$
$$= 0.08 \text{ mA}$$

With an output level of 0, the microprocessor must sink.

$$I_{INPUT} = 8 \times 1$$
$$= 8 \text{ mA}$$

*Table 2* summarises the microprocessor's output current requirements. (The minus signs indicate sinking currents.)

If the microprocessor output cannot source or sink the required currents, then buffer drivers must be added to boost the currents. *Figure 13* shows an example. The $I_{INPUT}$ requirements of the buffer are now 0.05 mA for the 1 level output of the microprocessor, and 1.6 mA for the 0 level. The buffer output can source or sink 30 to 40 mA, however.

# Glossary

| | |
|---|---|
| **bit length** | the number of bits a microprocessor is capable of processing at any time |
| **interface** | hardware required to connect a microprocessor and external functional devices |
| **interrupt** | process which allows a microprocessor to be diverted away from the task currently being performed to a new, generally more important, task |
| **microcomputer** | an integrated circuit which contains all the necessary blocks, e.g. microprocessor, memory, registers etc., to form a computer system. The name is often used, misleadingly, to define any computer system which contains a microprocessor |
| **output enable gate** | hardware feature on device outputs which allows the connection of more than one device to a common line or bus. An output enable gate has a three-state output |
| **output enable line** | the controlling line of an output enable gate. The logic state on this line determines whether or not the gate is active |
| **sink** | when a device accepts current from another device |
| **source** | when a device supplies current to another device |
| **steady-state current** | current on a data line or address line which occurs at all times |
| **three-state output** | output of a device which shares a common line, where the device may be electrically connected or disconnected. When connected, the output may be logic 1 or 0. The three states are thus: 1, 0 and disconnected |
| **transient current** | current which a device must be able to source or sink, when connected to a common data line or address line. The transient current occurs for only a short period of time |


# Optical systems–1

## Measuring light intensity

For any optical communications system it is vitally important to provide a definition of light **intensity**: the intensity of light is a measure of the energy contained in radiated light waves.

There are two main methods of measuring intensity: the **photometric** system, which is related to the human eye; and the **radiometric** system which is related to non-human sensors of light.

The photometric system is used to measure intensity in terms of the *sensation* of brightness perceived by the eye, so the system must take into account human perception factors. The radiometric system, on the other hand, measures light intensity purely in terms of energy, that is, its ability to do work.

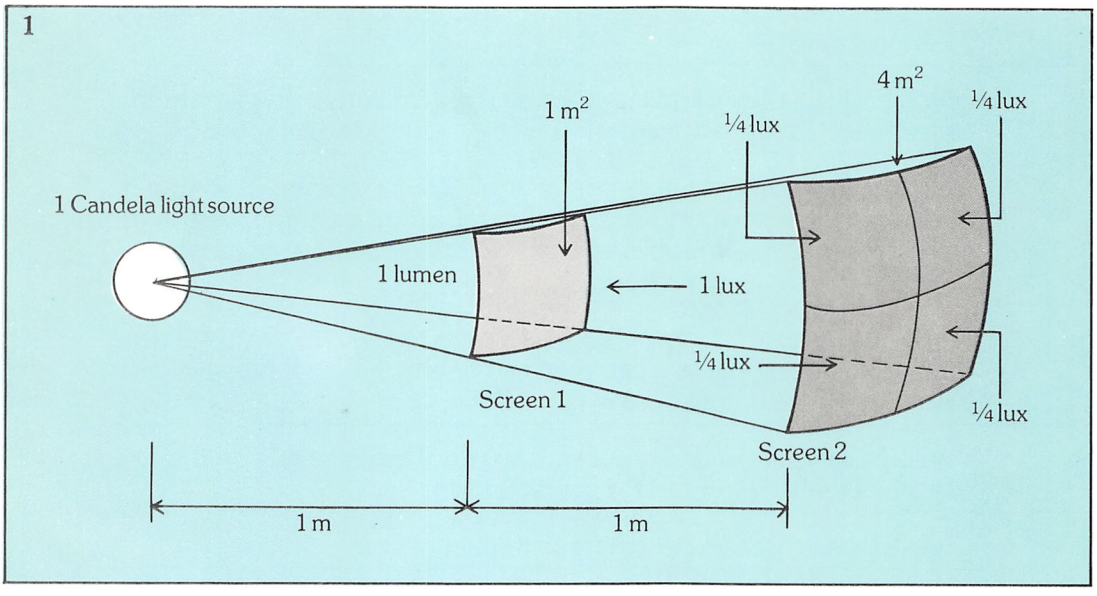
### Photometry

The photometric system for measuring intensity was developed to relate light measurements to the human eye because the human eye is a special type of sensor. Its function is to *see*, i.e. to determine the position, shape and colour of objects by converting an image in light to electrical impulses that are used by the brain.

Light is caused when electromagnetic waves, of wavelengths between about 10 $\mu$m and 100 nm, occur. Such wavelengths produce a spectrum of light from ultra-violet to infra-red.

Visible light, i.e. light which is detected by the eye, forms only a small part of the light spectrum. Wavelengths of visible light are between about 400 nm and 760 nm, producing colours of light from violet to red.

The eye does not respond equally to equal strengths of colours, however, and is in fact more sensitive to the colours in the middle of this range. By studying the eye's response to different strengths and colours of light, a graph known as the **visibility function**, or sometimes the **photopic response**, of the eye may be drawn. This shows that the human eye is more sensitive




**1. The intensity of light** falling on a surface varies inversely with the square of the distance between source and surface. A surface 1 $m^2$, receiving 1 lux illuminance at 1 m distance, will therefore receive ¼ lux illuminance when moved a further 1 m away.

to green light than it is to, say, violet or red. This means that, for example, a green light of lower intensity may produce the same effect as a violet or red light of higher intensity.

The following terms and units are used in the photometric system. The amount of light produced by a light source is called the **luminous intensity**. The standard unit used to measure luminous intensity is the **candela**. (For many years, the standard unit was the candle because the luminous intensity of a certain size *candle* made from the wax of sperm whales was used as the standard. The term *candlepower* has also been used to describe luminous intensity.)

One candela is the amount of light that shines through a hole in one side of a ceramic box after the box has been heated to 2042 K. The area of the hole is 1/60 $cm^2$ and the box is wrapped in platinum. It is heated until the platinum melts, then cooled until the platinum just begins to harden. The temperature at this point is exactly 2042 K, and the ceramic inside glows with intense light which shines through the hole in the box. The candela is used to calculate the lumen and lux which are the other units of light measurement.

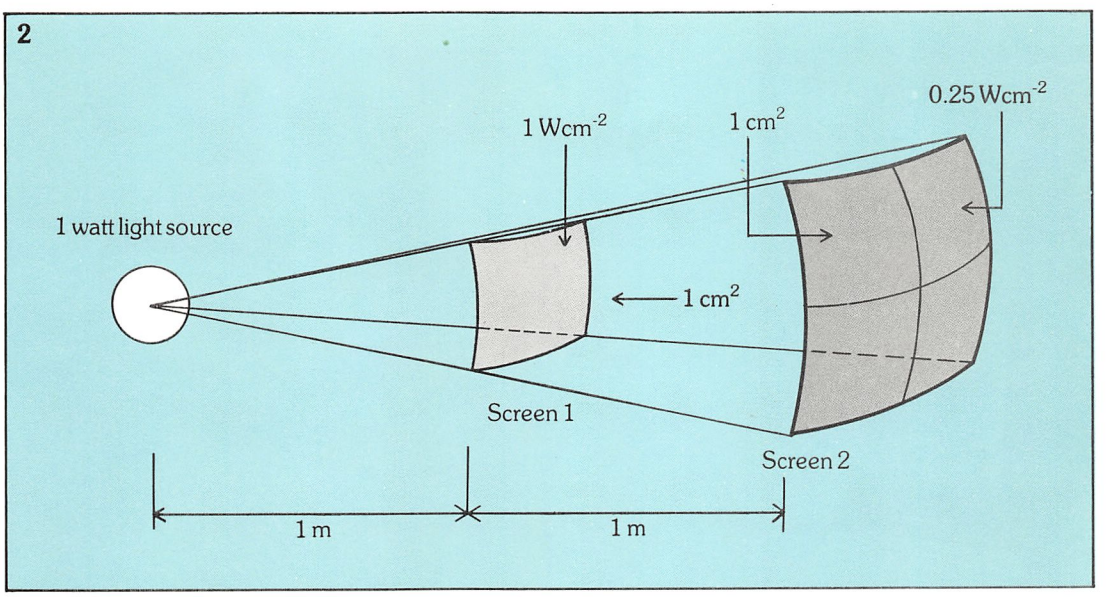
The **lumen** is used to measure the amount of energy in a beam of light. The **lux** is used to measure illuminance, that is, the amount of light shining on a surface. The relationship between the candela, lumen, and lux is: a 1 candela light source produces a 1 lumen beam of light which provides 1 lux of illumination on a 1 square metre area which is located on a radius of 1 metre from the source.

The intensity of light falling on a surface varies inversely with the square of the distance between the source and the surface. This means that if a surface receiving 1 lux illuminance at 1 metre is moved a further metre away from the surface, the surface will only receive 1/4 lux illuminance. This is illustrated in *figure 1*.

### Radiometry

We know that the visible spectrum is only a very small portion of the light spectrum which, in turn, is only a very small portion of the whole electromagnetic spectrum. Obviously, the human eye cannot be used to detect the presence of electromagnetic waves outside the visible spectrum – but other detectors, the non-human, man-made detectors can. For this reason, the radiometric system is used to measure the radiant energy of waves over the *whole* spectrum. Some of the important terms and units in the radiometric system are:

1) A **watt** is a unit of measure of the rate at which energy is radiated or used with respect to time. Sensors respond to this rate of change and convert a portion of the radiated electromagnetic energy to a useable form such as electrical, chemical or




**2. A 1W light source** is assumed to project 1 W of light uniformly onto a 1 $cm^2$ screen at 1 m distance, giving an intensity of 1 $Wcm^{-2}$. The intensity at screen 2, is only 0.25 $Wcm^{-2}$.

thermal energy.
2) The **intensity** of the radiation at a sensor is the ratio of the number of watts striking the sensor to the area of the sensor.
3) The **efficiency** of the sensor is the ratio of the number of watts converted to useable output to the number of watts striking the sensor.

These terms and units are illustrated in *figure 2*. The 1 W light source is assumed to project 1 W of light uniformly onto a 1 $cm^2$ screen which is 1 m away. The intensity at screen 1 is therefore 1 $Wcm^{-2}$. With screen 1 removed, the intensity at screen 2 is therefore only 0.25 $Wcm^{-2}$.

A sensor with an area of 0.01 $cm^2$ placed 2 m away from the source would receive 0.01 $cm^2 \times 0.25$ $Wcm^{-2}$, i.e. 0.0025 W of light, while the same sensor placed only 1 m away would receive 0.01 $cm^2 \times 1$ $Wcm^{-2}$, i.e. 0.01 W of light.

## The generation of light

The most common and earliest used mechanism to produce light was heat. The incandescent bulb, hot coals, fire, and gas lamps are examples of sources which use this technique. A substance is heated to a temperature sufficiently high so that the electrons in the atomic structure of the atoms are excited to higher energy orbits after which they decay to lower energy orbits.

Since each element has only a limited number of distinct orbits that it will allow the electron to occupy, light emitted from a particular atom will comprise only those wavelengths associated with that atom. This fact is often used to identify different compounds. For example, copper produces a green light.

The number of electrons and protons in an atom determines the type of material,


**Below:** laser scanning bar code reader built into a supermarket check-out. The laser scan (visible on the can at the right) is designed to interpret bar codes as items are quickly passed over the glass plate in the check-out bed.
(Photo: IBM).

as well as the electrical and chemical properties of that material. The orbits of the electrons are commonly called **shells** and these shells are arranged as shown in *figure 3*. Each shell has a number of electrons that it can hold. The K shell holds a maximum of two electrons, the L shell holds a maximum of eight electrons. The maximum number of electrons in the M and N shell is 18 and 32 respectively.

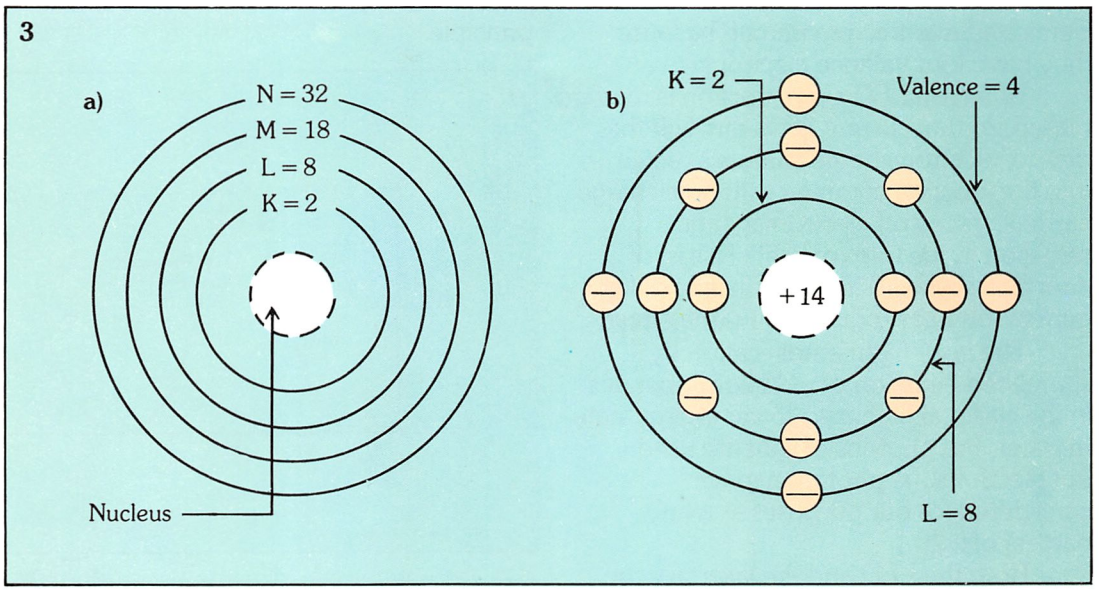
In addition to these maximum numbers, another necessary rule is that the outer shell of any atom has a maximum of eight electrons. In fact, the periodic table of elements used in chemistry and physics uses the number of electrons in this outer shell to define the groups of elements. This outer shell is called the **valence shell**.

If this valence shell contains only one electron, the material is a good conductor; that is, very small amounts of external energy are required to remove the electron from the outer shell and make it a 'free' electron which can move in the molecular structure of the material. Copper, gold, and silver, for example, have only one valence electron and are good conductors. The other important property of the valence shell is that if it is full (eight electrons), a large amount of external energy is required to pull an electron from the shell. Iron, cobalt, and platinum have eight valence electrons and are therefore poor conductors.

The forces in the atom are very complex. There are forces of attraction between the nucleus and the electrons, the forces of repulsion between the electrons in the various shells, and the forces due to the rapid motion of the electrons in their shells. The forces also depend on the neighbouring atoms.

As a result of these forces, there are discrete energy levels which can be associated with each shell. If the atomic structure is disturbed by adding energy in any form (heat, light, or electric field), the electrons may gain enough energy to move to a higher energy level within the atom. However, if the added energy is not high enough, the electrons will not change levels.

When an electron does change energy levels, the energy level which lost the electron needs an electron to complete its shell, and so develops forces to try to attract an electron. It may capture the original electron it lost or it may capture another electron. In either case when it does capture an electron, energy is released in the form of radiation. The wavelength of the radiation depends on the change in energy, which in turn depends on the atomic structure. In some materials, the radiation may be a relatively low frequency broad spectrum electrical noise. In other materials, it may be visible light. If visible, the colour of the light depends on the wavelength of the radiation. Some of the radiation may be




**3. (a) The orbits of electrons in an atom** are called shells and each shell can hold a maximum number of electrons; **(b)** the shell structure of silicon, showing the way that its electrons are arranged.

absorbed by the material itself and simply cause the temperature of the material to increase.

In any case, simply stated, when external energy is added, electrons are raised to a higher level; when electrons fall from the higher level to the lower level, light radiation is released.

### Optoelectronic light sources

In terms of optical communications systems the two most common light sources are the LED and the laser (see *Solid State Electronics 27*). The LED (light emitting diode) is a semiconductor device consisting of a p-n junction. Semiconductors are so named because they have four valence electrons.

When an LED's p-n junction is forward biased, so that current flows through the device, electron charge carriers injected into the junction combine with hole charge carriers, giving off electromagnetic radiation in the form of light. This is, of course, a slightly different principle of light generation but produces the same effect.

The laser (light amplification by stimulated emission of radiation) operates in the classical higher-to-lower energy state manner, but photons of light are used – not electrons. (A photon may be considered for our purposes as a tiny particle of light.)

Both the LED and the laser may be structured so that light of only one wavelength is generated. This monochromatic light is extremely useful, as will be explained in *Communications 10*. The laser has the extra advantage of producing coherent light, i.e. in phase.

### Detection of light

Light sensors used in electronics may be grouped into two broad categories: **thermal sensors** and **quantum sensors**.

Thermal sensors convert light energy to heat, which is in turn used to produce a voltage or a current. The thermocouple is an example of such a heat-to-electricity transducer, which produces a current proportional to the difference in temperatures of two junctions of dissimilar metals.

Quantum sensors are commonly divided into three subgroups:

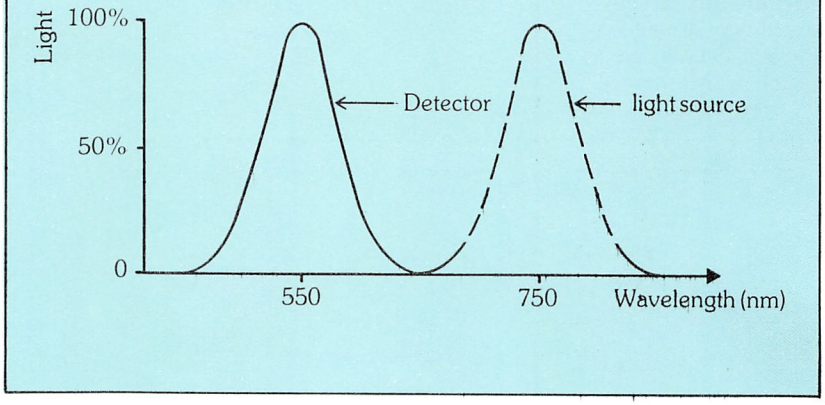
1) photoresistive sensors – which use light energy to control resistance;
2) photovoltaic sensors – which use light energy to produce an electrical voltage;
3) photoemissive sensors – which use light energy to free electrons from the sensor surface to produce a current.

The two most important sensors with respect to optical communications systems are the p-i-n diode and the avalanche photodiode (see *Solid State Electronics 22*). These will be covered in more detail in the following chapter, but for now it is sufficient to know that the p-i-n diode is a photovoltaic sensor and the avalanche photodiode operates on the photoemissive principle.

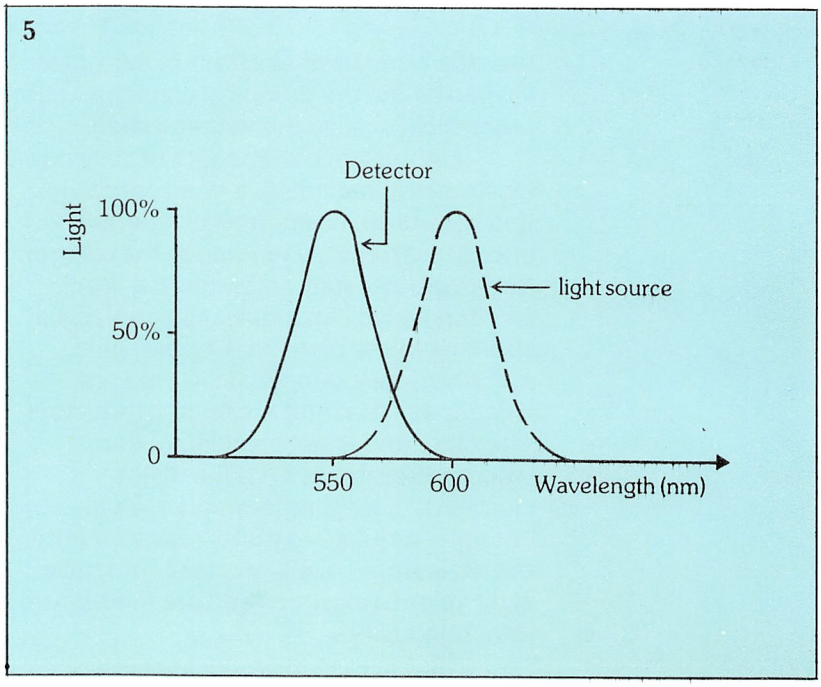
**4. If the wavelengths produced by a light source** do not overlap those to which a sensor responds, then these devices are said to be mismatched.

**5. If the wavelengths of the light source** and sensor overlap slightly, then the sensor will only give a fraction of its full response.

4




5



# Optically coupled electronic systems

Light sources and light sensors are always used together. An electronic system which uses a light source and sensor can be termed an **optically coupled system**.

Optically coupled electronic systems which we are interested in here may be generally divided into two categories – either **interruptible** or **non-interruptible**.

Interruptible systems are generally used to convey information about the medium between the source and sensor. Examples of interruptible systems are those used to count products on a conveyor belt, determine liquid levels, determine speeds and positions of objects, and read computer punched cards.

Non-interruptible systems, on the other hand, are used to pass information from the source to the sensor. Such systems can be used in a number of different ways, either simply to determine if the source is on or off, or in systems that have some unique feature – for example, to provide electrical isolation between source and detector, to provide a communication link to a computer, or to provide a wide bandwidth information channel.

For both interruptible and non-interruptible systems, similar considerations influence the design and construction of the systems. Let's now outline some of these.

**6. A perfectly matched** detector and light source.

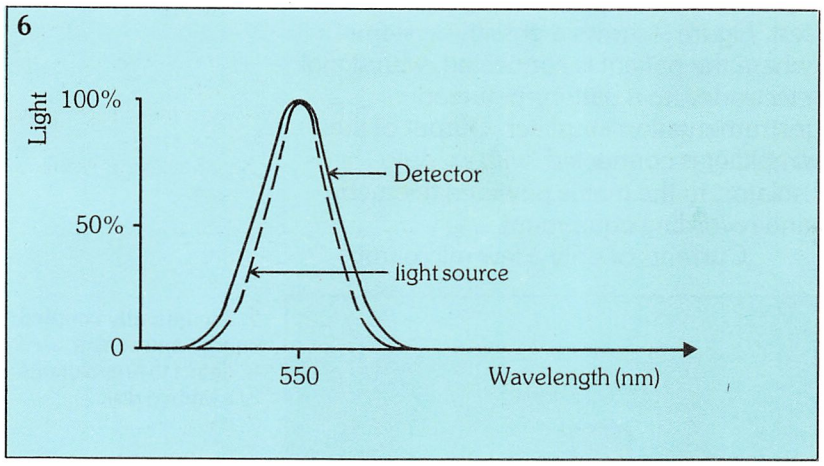



Table 1

**Typical matching factors for some common source–sensor pairs**

| Source | Sensor matching factor | |
|---|---|---|
| | Eye | Silicon junction |
| Sun | 0.16 | 0.5 |
| 2200 K tungsten lamp | 0.007 | 0.19 |
| 2600 K tungsten lamp | 0.021 | 0.24 |
| 3300 K tungsten lamp | 0.044 | 0.3 |
| Neon lamp | 0.35 | 0.7 |
| Gallium arsenide LED | 0 | 1.0 |
| Gallium phosphide LED | 0.08 | 0.7 |

## Matching

A very important consideration in optically coupled electronic systems is **matching** the spectral response of the light source and sensor. The compatibility of source and sensor can be determined by comparing the spectral responses of the source and sensor on the same graph, as in *figure 4*. If the wavelengths produced by the light source do not overlap the wavelengths to which the sensor responds, the two are said to be **mismatched**. The sensor can not 'see' the light at all. It is like transmitting a TV signal on channel 25 and expecting to receive it on channel 28. If there *is* some overlap, as in *figure 5*, the sensor would have some response to the source, but not as much as the pair shown in *figure 6*.

A matching factor can be defined by multiplying the sensor response curve and the source spectral curve, and measuring the area under the resulting curve. If the curves do not overlap at all, as in *figure 4*, the area will be zero. This means that the sensor will not absorb any energy from the source. If the curves totally overlap as shown in *figure 6*, the area is unity, and the sensor would absorb all of the energy from the source which falls on it. *Figure 5* is in between the two extremes of *figures 4* and *6*. Here only a portion of the energy radiated by the source is absorbed by the sensor.

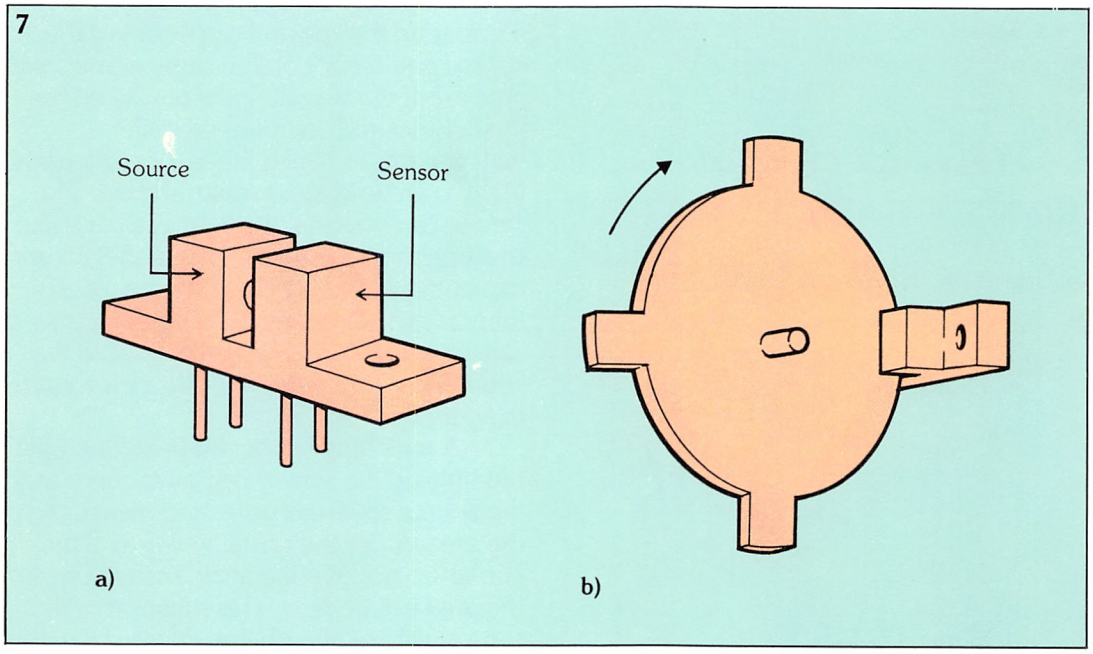
*Table 1* lists typical matching factors for a few common source-sensor pairs. Two sensors are shown: the eye, and a sensor made from a silicon semiconductor junction. Examining these factors, we can see that the silicon sensors are better than the eye for all sources listed.

## Transmission media

Another consideration of importance in optically coupled electronic systems is the medium between source and sensor. Air is a commonly used medium and an example of an interruptible optically coupled source and sensor with an air medium is shown in *figure 7a*, as a module which houses them. Between the source and sensor is a small gap (about 3 mm) of air. *Figure 7b* shows how such a module may be used with a tabbed disk to detect revolutions. As the disk turns each tab interrupts the light transmission between source and sensor. The frequency or rate of the interruptions indicates how fast the disk revolves. Source/sensor modules like the one in *figure 7* are very common.

Electrical isolation properties of the opto-isolator are useful in a number of applications. In a system, for example, which uses high-voltages (for power) but which must be interfaced to humans, opto-isolators may be used to prevent harmful current flow. Medical instrumentation equipment such as electrocardiograph or electroencephalograph machines use opto-isolators to protect the patient under test. *Figure 9* shows a possible system where the patient is connected, with signal electrodes, to a battery-powered instrumentation amplifier. Output of the amplifier is connected, with an opto-isolator, to the mains powered telemetry and recording equipment.

Currents of only a few microamps may be dangerous at a high enough voltage, especially if they flow near the heart. The battery powered instrumentation amplifier avoids this problem, while the opto-isolator allows safe connection to the high voltage equipment.




**7. An optically coupled source and sensor** used to detect the revolutions of a tabbed disk.

### Electrical isolation

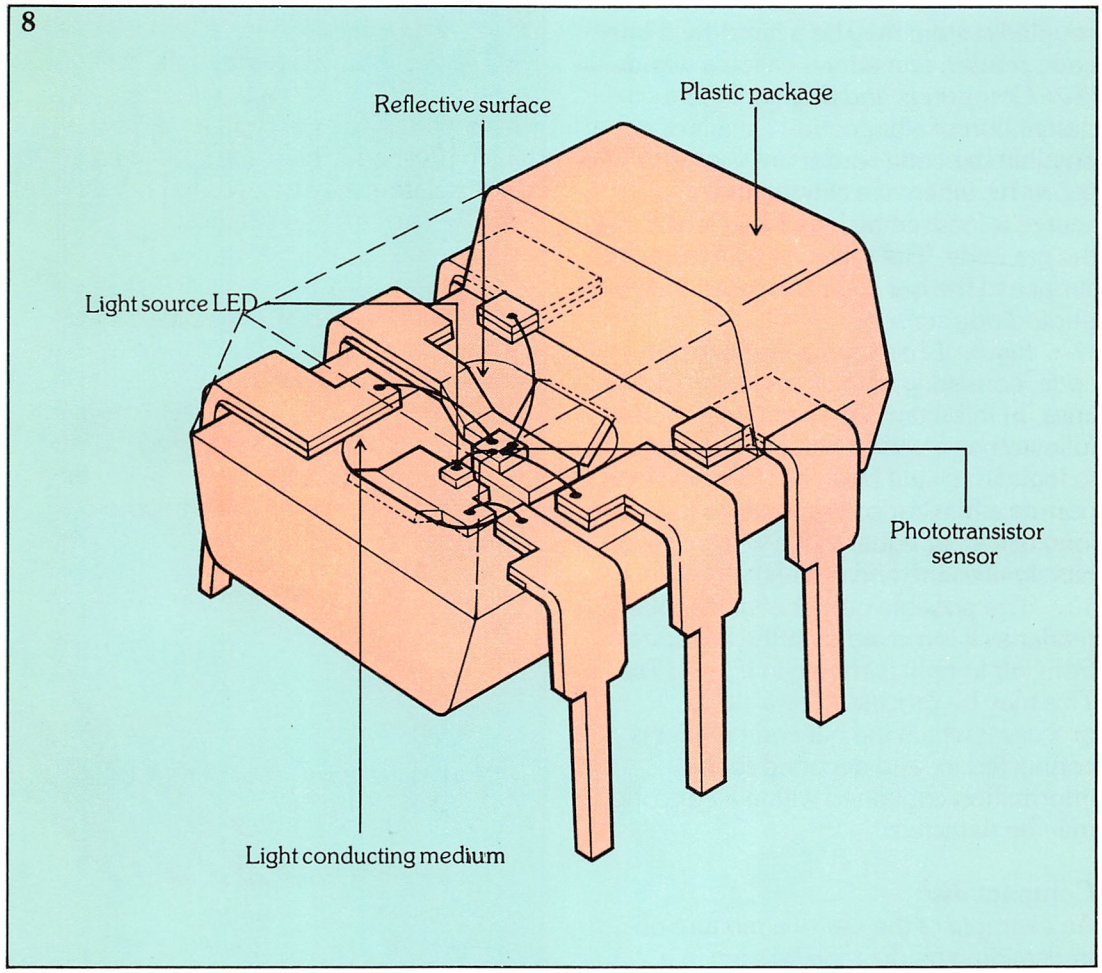
*Figure 8* shows a non-interruptible optically coupled system. It is known as an **optical isolator**, sometimes abbreviated to **opto-isolator** or **opto-coupler**.

The source (an infra-red LED) and sensor (a phototransistor) are mounted together in an integrated circuit type package, electrically isolated from each other. They are separated by a medium which is transparent to the frequencies generated by the sensor and detected by the sensor, but which is an electrical insulator.

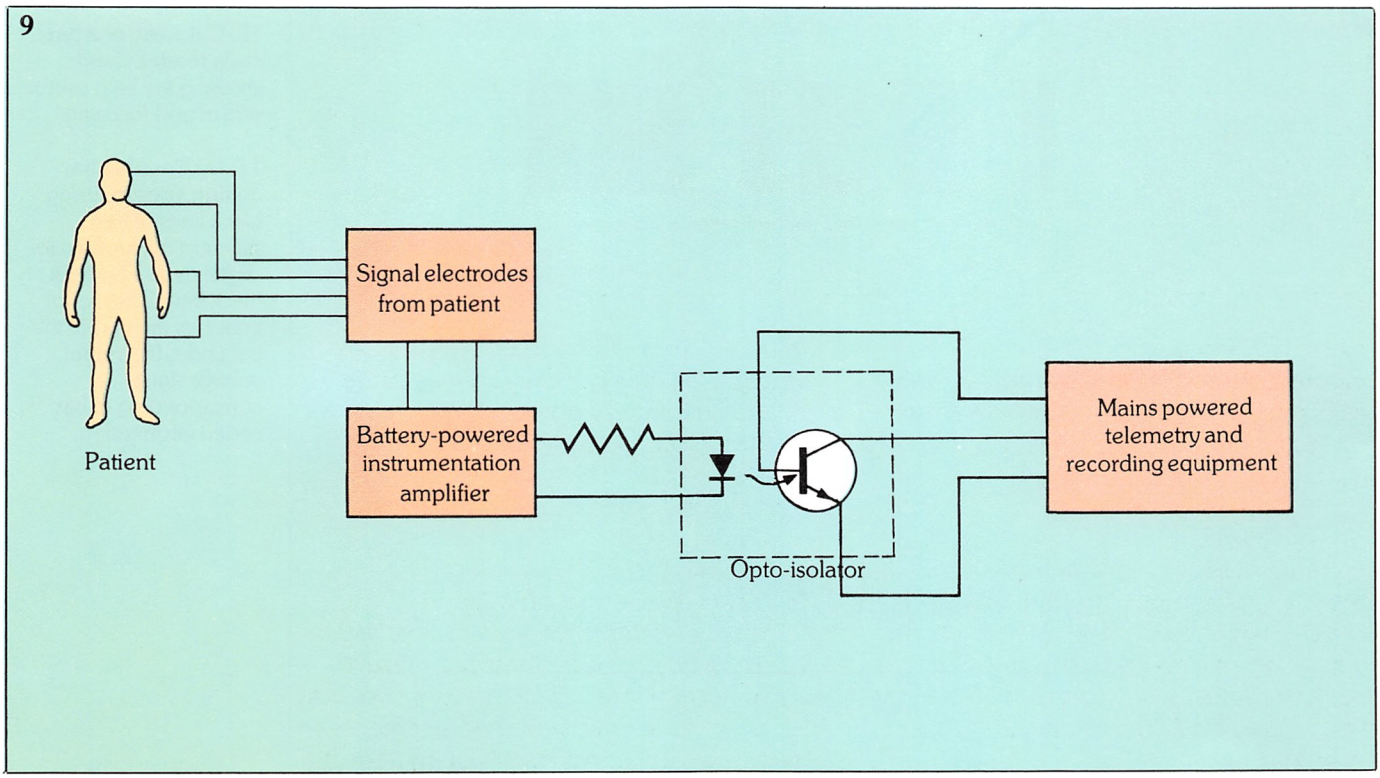
### Mirrors, lenses and prisms

Other media regularly used in optically coupled electronic systems are mirrors, lenses and prisms, or mixtures of two or all of these. An example of the use of lenses as the transmitting medium of an optically

**8. Cutaway of a non-interruptible optically coupled systems** – otherwise known as an opto-isolator.




**9. Opto-isolator in use**, employed here to protect a patient under examination from high voltages.



coupled system may be a hand-held **bar code reader**, sometimes called a **wand**. (See *Computers and Society 9* for a description of a bar code.) Details of a possible bar code reader are shown in *figure 10*, where we can see that a LED source is focused by lens 1 and lens 2 onto the bar code. The reflected light from the surface of the bar code is focused onto a photodiode sensor.

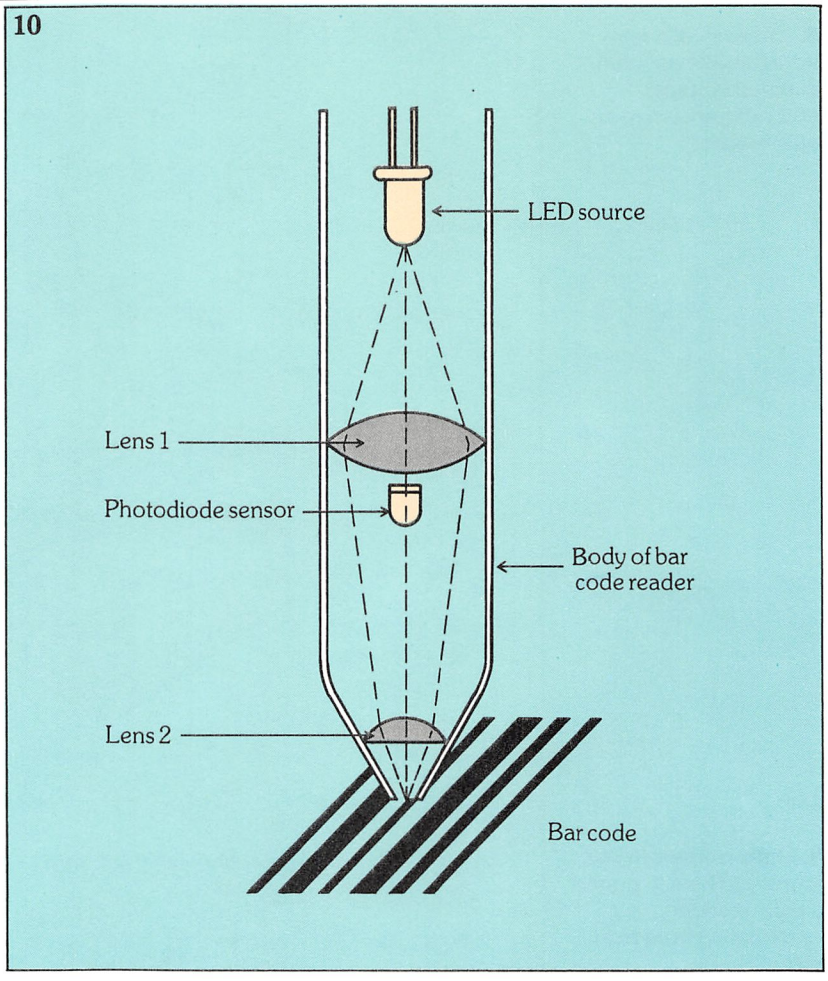
*Figure 11a* shows a possible bar code, consisting of thick and thin black lines. In this example a narrow black bar followed by a wide white bar corresponds to logic 0. A wide black bar followed by a narrow white bar corresponds to logic 1. A long black bar equal to the width of two bits signifies the end of the bar code.

The possible output of a bar code reader as it is moved over the bar code from left to right is shown in *figure 11b*. This may be processed by a signal processor which the bar code reader is connected to, and decoded so that information contained within a bar code may be detected.

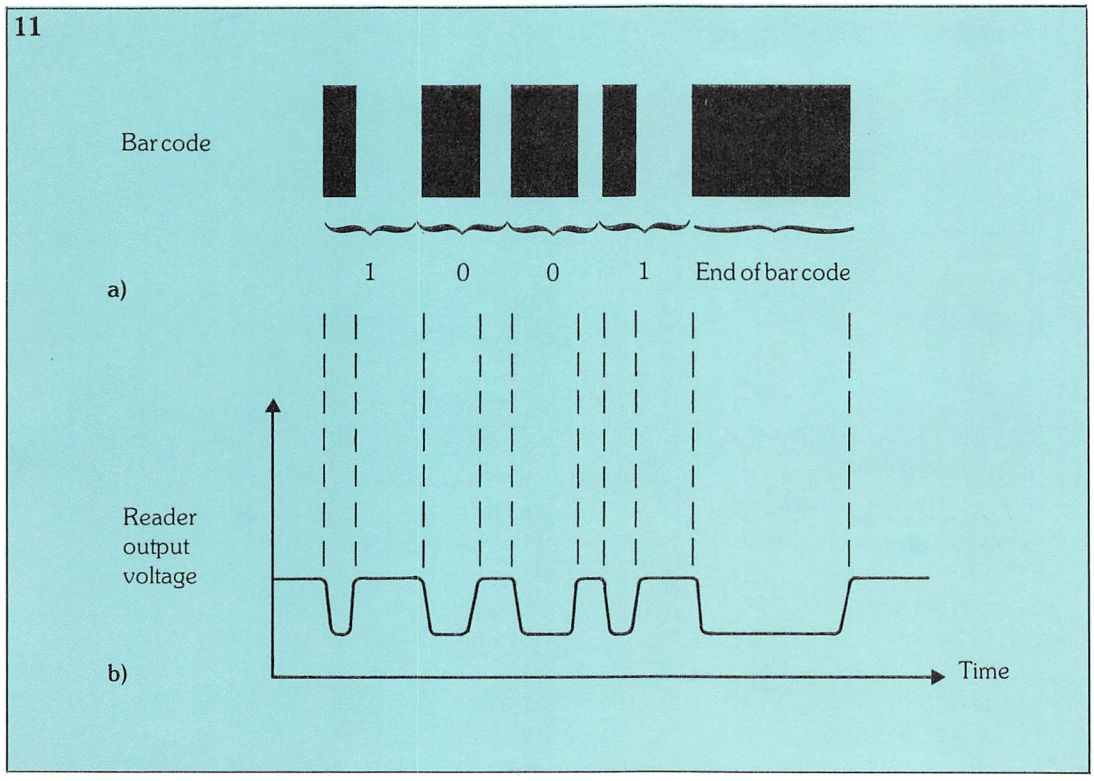
## Compact disc

An example of the use of a mixture of transmitting media may be seen in a

10




11




**10. Cutaway of a bar code reader wand**, showing the light source, sensor and focusing.

**11. (a) Possible bar coding system**, using black lines of two different thicknesses for digital data. The thick black line (2 bits long) indicates the end of the bar code; **(b)** output voltage signal corresponding to bar coded information.

compact disc digital audio player. The principle of operation of a compact disc player is that light from a laser source is focused onto the internal surface of a digitally recorded disc. Light reflected from the surface is then focused onto a photodiode. The output voltage of the photodiode is then ready for signal processing, decoding, amplification and replay through a loudspeaker as sound.

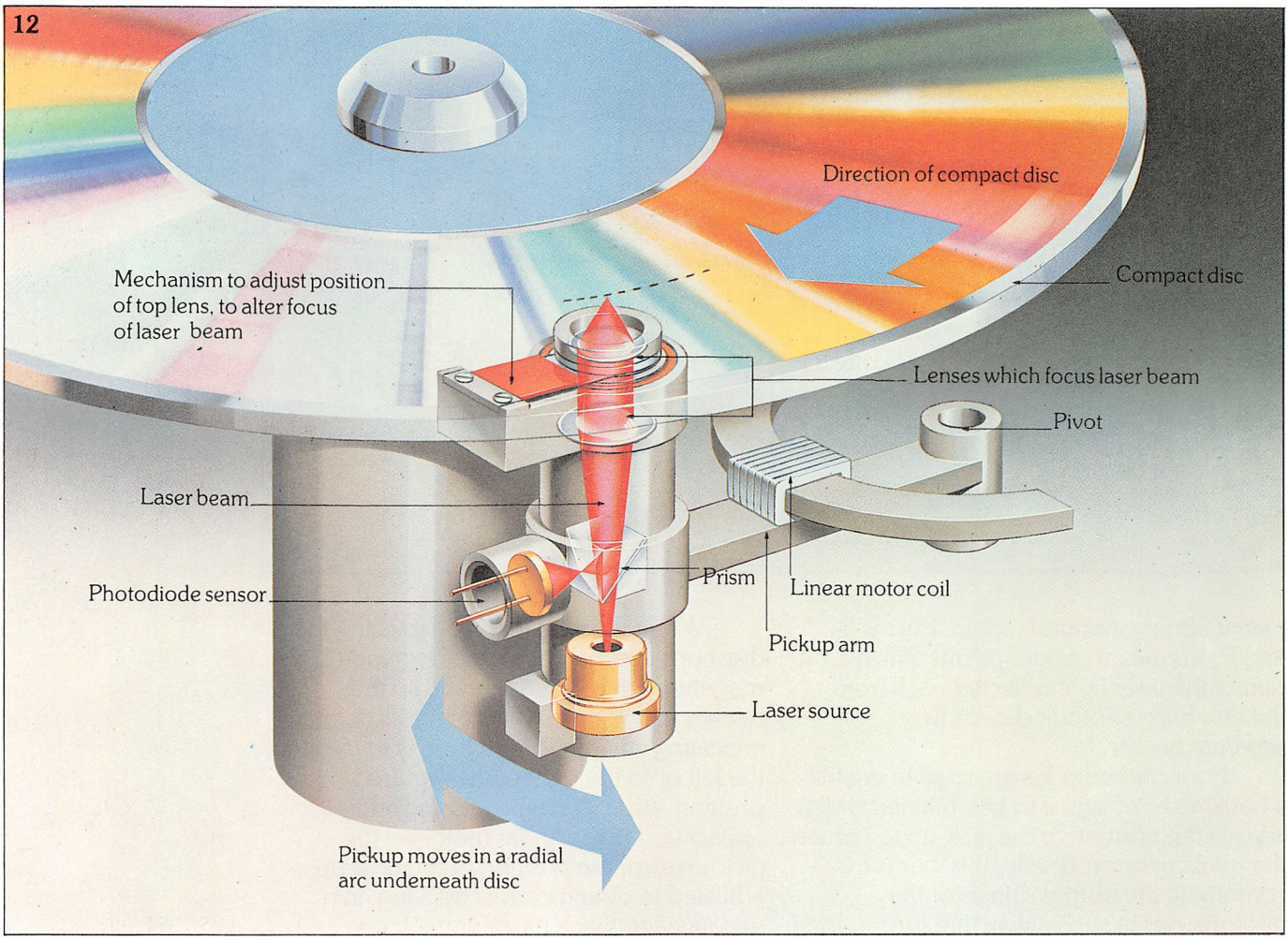
*Figure 12* shows the principle of operation of a compact disc player mechanism. The optical pick-up is mounted on an arm which is pivotted so that the pick-up can freely move across the underside of a compact disc, describing a radial arc across the disc, allowing it to scan the complete recorded information.

Close to the pivottal point of the optical pick-up arm is mounted a linear motor, comprising a combination of a coil and a permanent magnet. When the coil is energised the pick-up can be directed to any required part of the disc, the locational information being provided by some of the information actually recorded in the disc itself. The pick-up mechanism is therefore able to find independently any passage of music required by the user.

Light from the laser source shines up through the prism in the centre of the pick-up and is focused with two lenses, to a spot within the compact disc. Reflected light from the disc passes back through the two lenses but is refracted by the prism out to the photodiode.

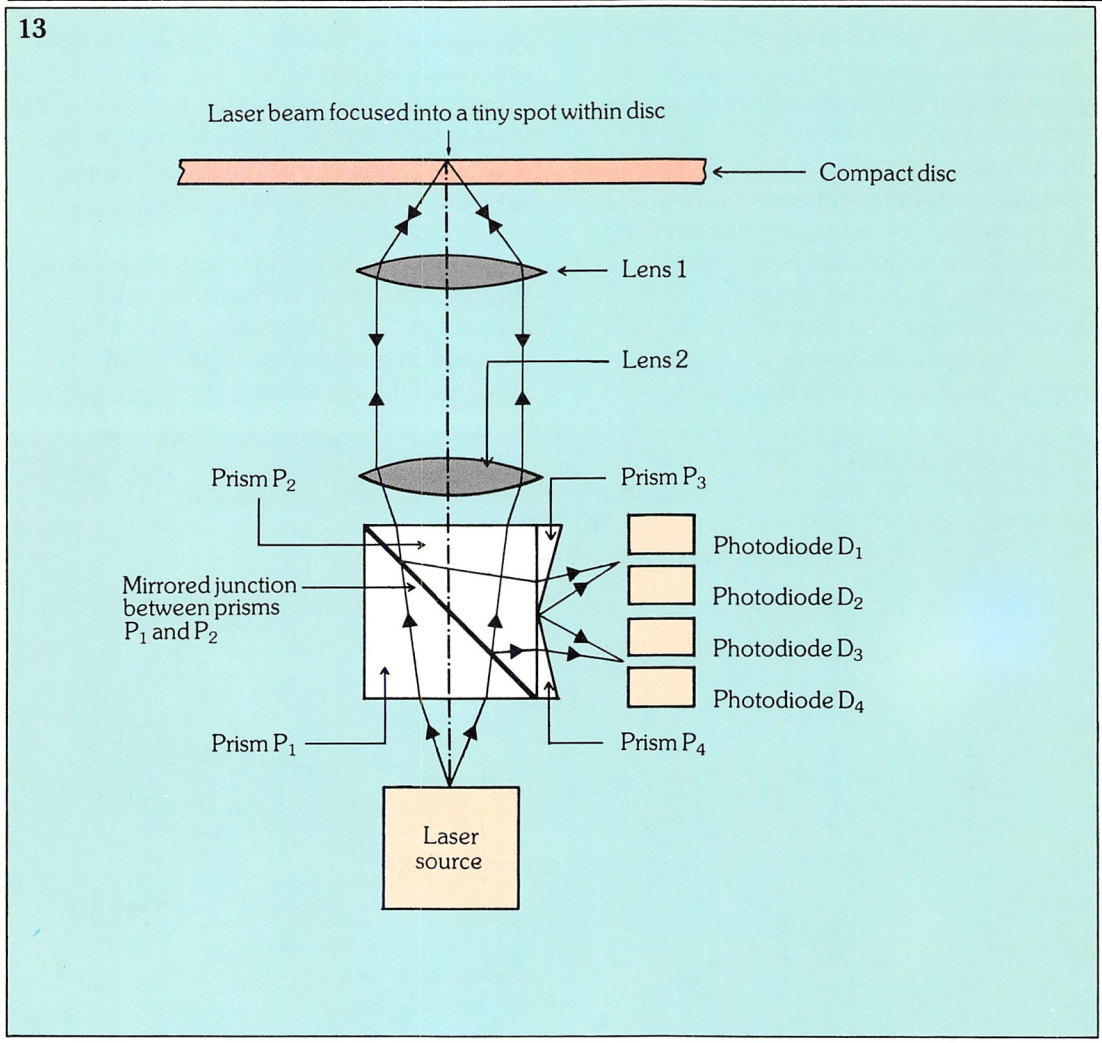
Although correct in principle, this description is simplified. *Figure 13*, however, shows the main optical parts of the pickup in greater detail.

The prism is made up of four smaller prisms. Prisms $P_1$ and $P_2$ are joined together to form a cube, after one of the two joining surfaces has had a half-silvered




The Research House/Philips

**12. Compact disc player mechanism.** The system uses an optical pick-up which moves in an arc underneath the rotating disc, driven by a linear motor.




**13. The main optical parts** of the compact disc pick-up, shown in detail.

mirror film evaporated onto it. Prisms $P_3$ and $P_4$ are beam-splitter prisms which refract the laser light reflected back from the disc onto photodiodes, as two separate beams.

Four photodiodes are used to enable a tracking error signal to be obtained which adjusts the position of the pick-up to reduce the tracking error. Briefly, this works by combining the output signals of the photodiodes in such a way that the magnitudes of the signals correspond to the magnitudes of the two beams of laser light. When the pick-up tracks the disc correctly, the two beams are of equal magnitude. The **tracking error signal** is given by:

$(V_{D1} + V_{D2}) - (V_{D3} + V_{D4})$

so when the beams are of equal magnitude, the output voltages of all photodiodes are equal and the tracking error signal is zero.

When the pick-up is not tracking the disc correctly, one beam has a greater magnitude than the other and so the tracking error signal is not zero, its polarity indicating whether the pick-up is tracking to the left or to the right of the required position. After signal processing and application to the linear motor on the pick-up arm, the position of the pick-up is adjusted to obtain correct tracking, in a negative feedback type control loop.

As a result of ageing or soiling of the optical system, the reflected beam may acquire a slowly increasing, more or less constant asymmetry. This creates a DC component in the tracking error signal, which causes a constant mis-tracking to occur. To compensate for this effect a second tracking error signal is generated.

To do this, the coil that controls the pick-up arm is supplied with a small alternating voltage at 600 Hz. The output

sum signal from all four photodiodes, i.e:

$V_{D1} + V_{D2} + V_{D3} + V_{D4}$

which is at a maximum when the pick-up is tracking correctly, is thus modulated by an alternating voltage of 600 Hz. The amplitude of this 600 Hz signal increases proportional to the amount the pickup mis-tracks. So a DC voltage equal in magnitude but opposite in polarity to the DC tracking error signal may be derived to counteract this mis-tracking.

Outputs of the four photodiodes are used in another way to control focussing of the laser beam onto the disc. As the disc itself rotates, any deviations in the distance between it and lens 2, mean that the beam will not be correctly in focus. To counteract this, lens 2 is mounted in a mechanical arrangement consisting of a coil and a permanent magnet – the lens may then be moved vertically up or down, much like the cone of a loudspeaker, depending on the current through the coil.

If the laser beam is sharply focused onto the disc, two sharp images are precisely located between photodiodes $D_1$ and $D_2$, and between $D_3$ and $D_4$. If the beam is not sharply focused, however, the two images on the photodiodes are not sharp either, and will move closer together or further apart. The signal obtained by connecting the photodiodes' output voltages as:

$(V_{D1} + V_{D4}) - (V_{D2} + V_{D3})$

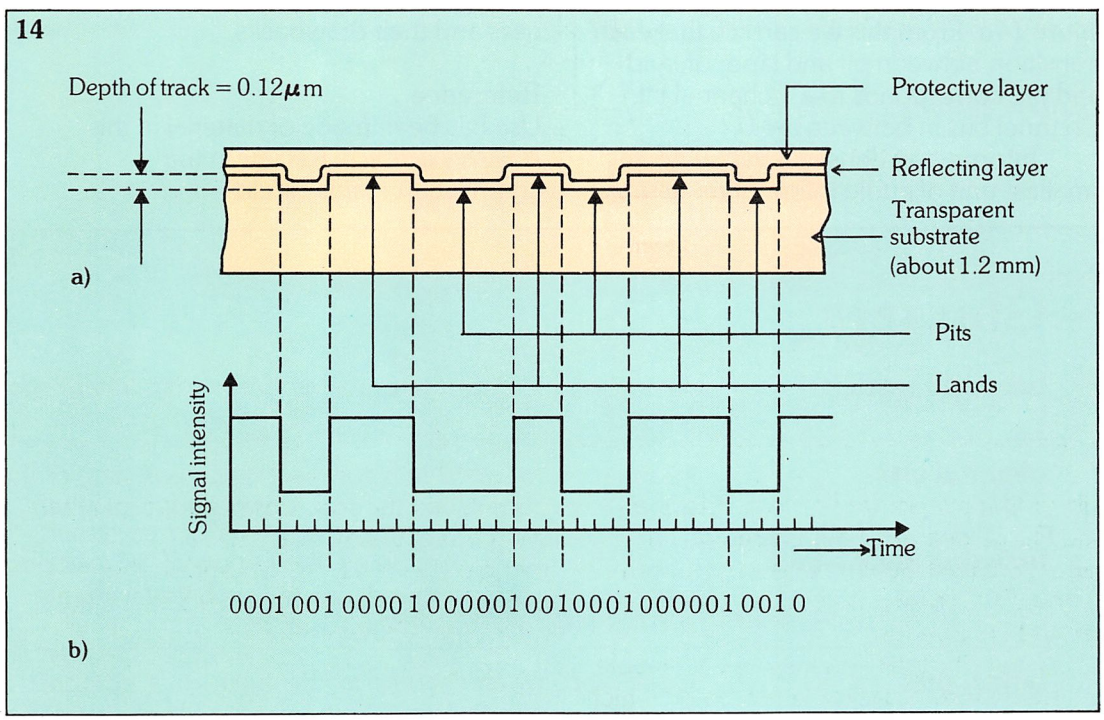
is therefore a **focusing error signal** which is suitably processed to drive the coil, to reposition lens 1 and refocus the laser beam.

### The disc itself

Pulse code modulation is used to convert the analogue music signal into a digital signal for the purpose of recording. We know from the sampling rule that the frequency at which samples must be taken should be at least twice the highest frequency in the music signal. We also know that the number of bits in each sample determines the quantization error and hence the quality of the recreated music signal.

A sampling frequency of 44.1 kHz into 16 bits per channel (i.e. 32 bits for the stereo signal) words is used which allows recreation of a music signal up to and over 20 kHz, with a quantization error of less than 0.002%. A quantization error as low as this corresponds to a signal-to-noise ratio of the recreated signal of over 90 dB.

Before recording into the disc, this pulse code modulated signal has extra bits added, in a coding system known as




**14. (a) Cross-section of a compact disc**, showing the pits and lands; **(b)** high and low signal intensities defined by the pits and lands, correspond to a 1 channel bit at each transition.

**cross-interleaved Reed-Solomon code** (CIRC) which allows error correction of the recreated signal. Also added are **control and display** (C and D) bits which contain information relating to playing time, composer, time etc., as well as the locational information mentioned earlier.

This bit stream is then modulated into a single spiral track in the disc, comprising a succession of **pits**. Intervals between the pits are known as **lands**. Each pit and land represent a series of bits called **channel bits**. *Figure 14a* shows a cross-section of a disc where the pits, lands and thicknesses of relevant layers are marked.

*Figure 14b* shows a graph of signal intensity read by the optical pick-up, corresponding to the section of track in *figure 14a*. From this we can see that each transition between pit and land, or land and pit, corresponds to a 1 channel bit. Channel bits in between are 0.

Information density is very high: the smallest unit of audio information (a single bit) covers an area of only $1\mu m^2$; track width is only $0.6\mu m$, and consequently the diameter of the laser beam spot must be only about $1\mu m$.

Unlike the common vinyl record which is tracked by a mechanical pick-up at a constant **rotational speed** (i.e. 45 rpm or 33 rpm), the compact disc is tracked by its laser pick-up at a constant **linear velocity** of $1.25\ ms^{-1}$. This means that the compact disc's speed of rotation must vary with the pick-up's position.

### Fibre optics

The most important of all optical transmission media is optical fibre. *Communications 10* will explain in detail the types of optical fibres available, their uses and their drawbacks.

### Reference

Use has been made of material in the *Philips Technical Review* in the compilation of this article.

## Glossary

| | |
|---|---|
| **bar code reader** | device used to detect information held within a bar code. Often called a wand |
| **channel bits** | information recorded in a compact disc, consisting of the audio music signal, error correction bits, and control and display bits |
| **focusing error signal** | signal obtained by combining the outputs of the four photo diodes of a compact disc player optical pick-up, such that it equals: $(V_{D1} + V_{D4}) - (V_{D2} + V_{D3})$ |
| **lands, pits** | raised and lowered areas in the track of a compact disc. Distance between a pit and a land (i.e. the depth of track) is only $0.12\ \mu m$ |
| **optical isolator, opto-isolator, opto-coupler** | optical source and sensor mounted together in a single integrated circuit type package, but electrically isolated |
| **photometry** | method of measuring light, related to the human eye |
| **radiometry** | method of measuring light, related to purely non-human sensors |
| **tracking error signal** | signal generated when the optical pick-up of a compact disc player mis-tracks. Formed by combining the outputs of the four photodiodes so that it is: $(V_{D1} + V_{D2}) - (V_{D3} + V_{D4})$ |

# Control signals

**Below:** modern 'communications receiver', that utilises a microprocessor controlled tuning section. This holds a list of memorised frequencies and tunes into stations automatically. (Photo: CBC Italy).

## Address decoding

*Figure 1a* shows a 1024 words × 10 bits memory, being fed from a microprocessor's address bus. As you can see, only 10 out of the 16 address lines are being utilized – only 1024 ($2^{10}$) words out of a possible total of 65,536 ($2^{16}$)!

*Figure 1b* shows how the memory's capacity has been partially expanded to 8192 ($2^{13}$) words, by connecting a 3-line to 8-line decoder to three of the 'spare' address lines. The $\overline{E}$ signal shown coming from the decoder to each memory block in the diagram is a **memory enable**. When $\overline{E}$ is at 0, then the corresponding memory block is enabled. So, each of the eight, 1024 word x N bit memory blocks can be enabled at selected times by controlling the $\overline{E}$ signal.

The third block down in *figure 1* can be seen to be enabled, as it is receiving a 0 enable signal and all the others are set at level 1. The 3-line to 8-line decoder circuit achieves this by taking the three most significant bits ($A_{10}$, $A_{11}$ and $A_{12}$) beyond the first ten bits ($A_0$ to $A_9$, which are being fed to each of the 1024 bit memory blocks). The $A_{10}$, $A_{11}$ and $A_{12}$ bits thus decide which of the eight memory blocks receives the 0 enable signal, while the bits $A_0$ to $A_9$ address the individual memory locations. We can see that 8 × 1024 = 8192 words of memory space are now available, through the use of additional decoder circuitry.

As we saw in earlier *Digital Electronics* chapters, 3-line to 8-line decoders are standard ICs, and we could expand the memory up to its full capability of 65,335 words with the addition of further decoders. By using the right combination of memory circuits and decoders any size of memory can be built – up to the number of words that the microprocessor's address code can handle.

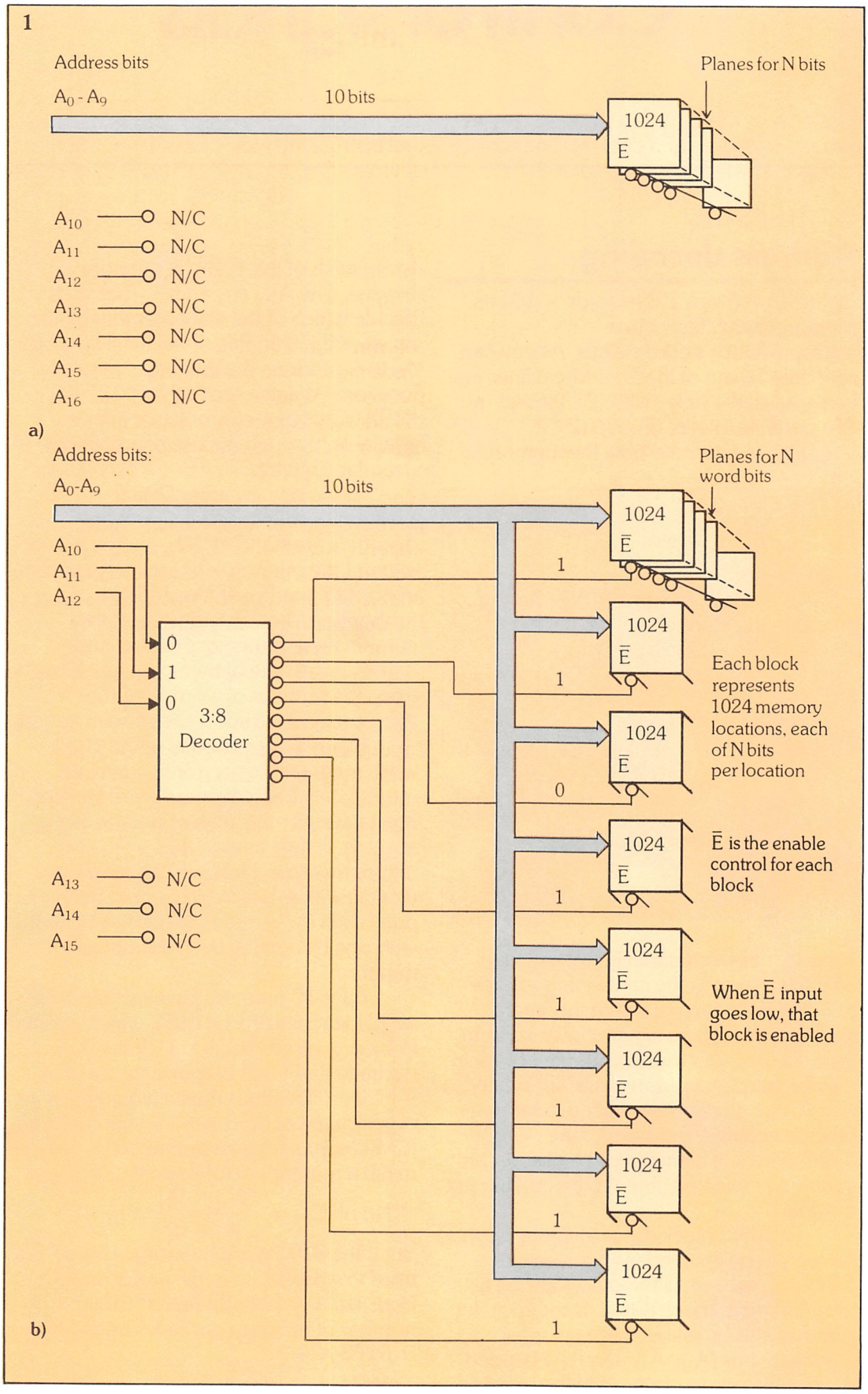
*Figure 2* looks at one of the 1024 word memory blocks from *figure 1*, in more detail. Each memory word location must have as many bits of storage as the number of data lines in the microprocessor. Here we can see we are dealing with 4-bits (which would be the number N in *figure 1*) making a 4-bit word with one bit coming from each block of memory.

Each IC memory package has 1024 word locations of 1 bit each, which are said to be organised '1024 × 1'. If the IC package had 1024 bits, arranged as 256, 4-bit word locations then it would have an organisation of '256 × 4'. The number of IC memory packages organised $W_I \times B_I$ to make a memory of total size $W_M \times B_N$ is:

$$\text{No of ICs} = \frac{W_M \times B_N}{W_I \times B_I}$$

So, if the 8192 word memory in *figure 1* is used to store 4-bit words, and is made from 1024 × 1 bit ICs then the total number of memory packages needed is:

$$\frac{8192 \times 4}{1024 \times 1} = 32$$




**1. (a) A 1024 word memory** is fed from 10 of a microprocessor's 16 address lines; **(b)** using only three more address lines partially expands the memory capacity to 8192 words.

As you can imagine, each memory block would be made up of four IC packages. *Figure 2* shows that each individual package would have to have 10 bits of the address code connected to it, as well as the data-in and data-out line for each bit. The $\overline{E}$ signal from *figure 1* is now connected to the $\overline{CE}$ terminal on each chip. The $\overline{CE}$ stands for chip enable or chip select, so when the $\overline{E}$ line is active each $\overline{CE}$ connection activates a 1024 × 1 memory package. This ensures that all four stored bits in each word are activated at the same address, so they are written or read at the same time.

So far we have only been discussing RAM, which could be used for program or data storage. The same decoding principles do apply to ROM of course, but the data-in lines and write signals would not be needed.

The number of input and output units that may be connected to a microprocessor can also be increased by using decoding circuits in the manner that we have discussed. The decoder units would control the additional input/output units with enable signals as we have seen them do with memory. The address lines must be distributed to the additional input or output blocks, and the data input and output lines gated onto the data bus using the circuits that we looked at earlier on.

## Timing and control line connections

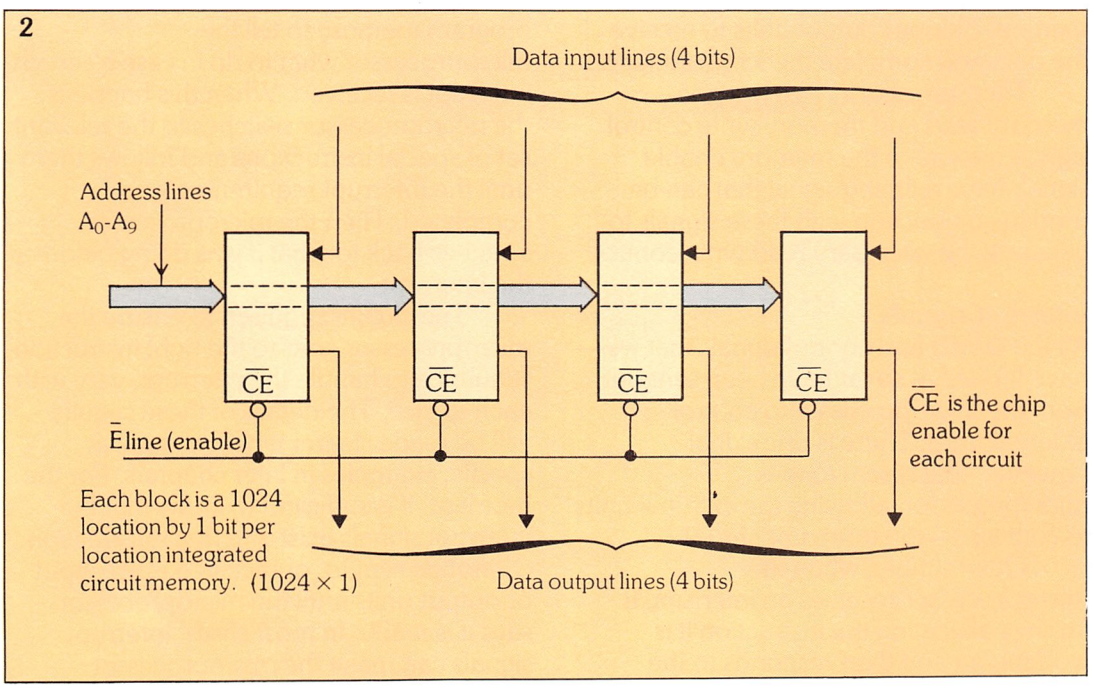
The microprocessor must generate signals that instruct the memory and input/output devices when they are to be turned on and whether they are being read from or written into, to keep the system's operation orderly. The following signals are used to control the operation of the external components:

### Memory enable signal

Once the microprocessor knows that a memory operation is to be performed, and the address signals are on the address lines, it must turn on the memory and tell it to read or write. Some microprocessors send out both signals, as *figure 3* shows. A memory enable signal turns the memory on, and a read/write (R/W) signal tells it to read or write data. Both of these signals enable the memory, so that it carries out the operation in the desired way. The memory enable signals must of course last long enough for the read or write operation to be completed.

### Input/output enable signal

Some microprocessors treat the input/output devices as a system separate from the memory. In this case (*figure 4*), the




**2. One of the 1024 word memory blocks** from *figure 1*, shown in greater detail.

microprocessor sends out an I/O enable signal. A read/write signal is also employed in the same way as we have just seen it used to access memory.

It is also possible that the microprocessor could send two separate control signals – an I/O read and an I/O write. These allow the input/output blocks to share the same address codes as parts of memory, as the memory and the I/O devices would not be enabled at the same time. Microprocessors that don't have a separate input/output system assume that these devices are assigned to memory locations. That's to say they are treated as if they are a part of memory. This method is called **memory mapped I/O**, since a certain number of memory locations are assigned to the system's input and output devices and reserved for their use.

### Read/write control signals

As we know, read/write control signals instruct the external functional blocks whether they are to send information *to* the microprocessor (read operation), or receive information *from* the microprocessor (write operation). The signal is normally held in the READ state, and only enters the WRITE state for a brief time, during which the microprocessor places data on the data lines. The WRITE state remains on long enough for the memory or input/output units to receive the data and complete the write operation.

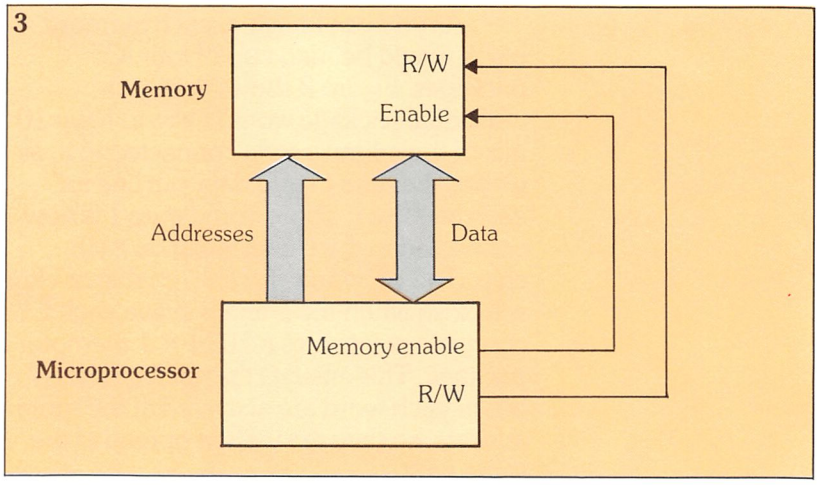
Microprocessors that provide memory read and memory write control signals instead of the memory enable signal, have a timing signal that can be used in conjunction with these signals to generate the necessary read/write control.

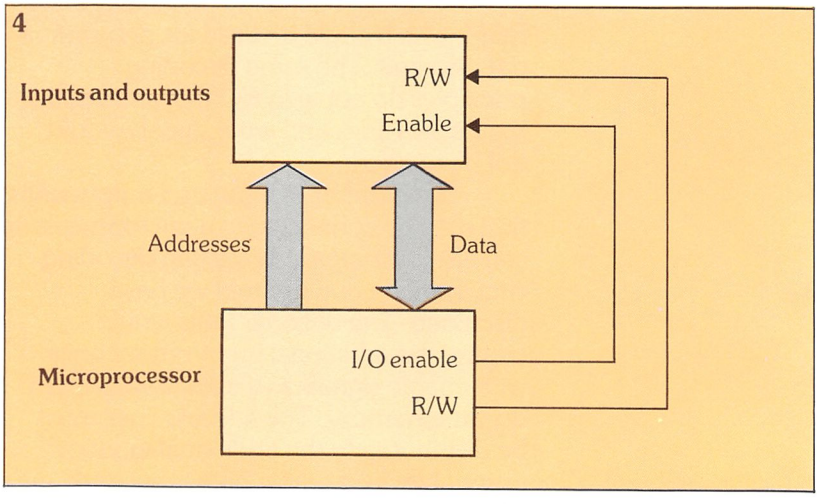
### Interrupt signals

All the timing and control signals that we have looked at so far have been sent out *from* the microprocessor to control external units. There is one signal however, that is sent *to* the microprocessor, allowing the external units to control it – the **interrupt signal**.

Quite simply, when the microprocessor receives an interrupt, it finishes executing the instruction it is working on and then responds to the interrupt – as *figure 5* shows. Special sets of instructions have been set up in the program memory to tell the microprocessor what to do in case a certain interrupt is received. When this happens, the microprocessor switches to the relevant set of special instructions and follows them until the interrupt requirements are completed. Then the microprocessor switches back to what it was doing before the interrupt.

The circuits required to ensure the microprocessor goes to the right instruction sequence to handle the interrupt, vary with each device. The details of these circuits will be made clearer when we look at specific examples in later chapters. For the moment, it is sufficient to know that the interrupt signals exist and that information cannot be sent to, or received from, input or output units until the microprocessor says it is ready. In most cases, interrupt signals can make the microprocessor

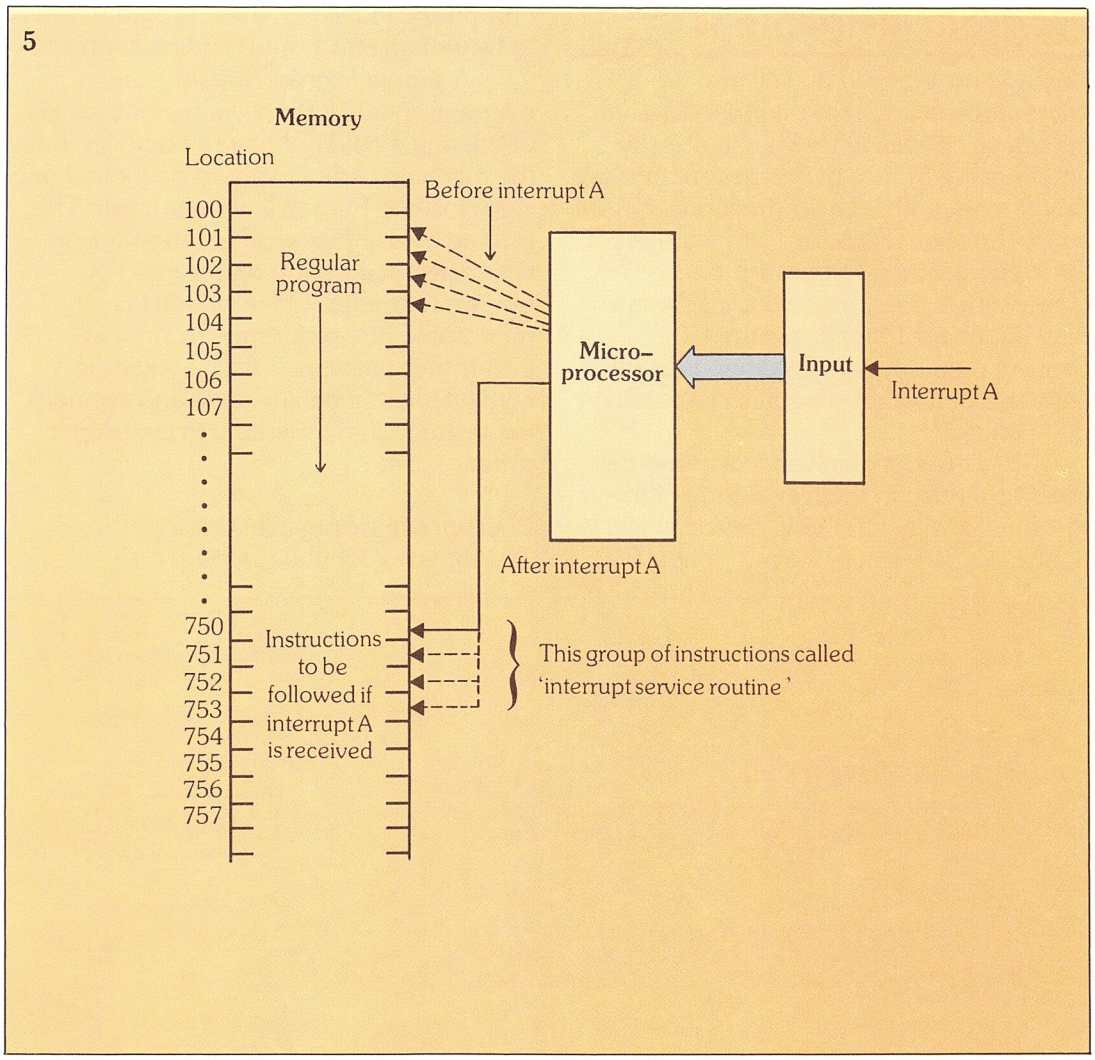


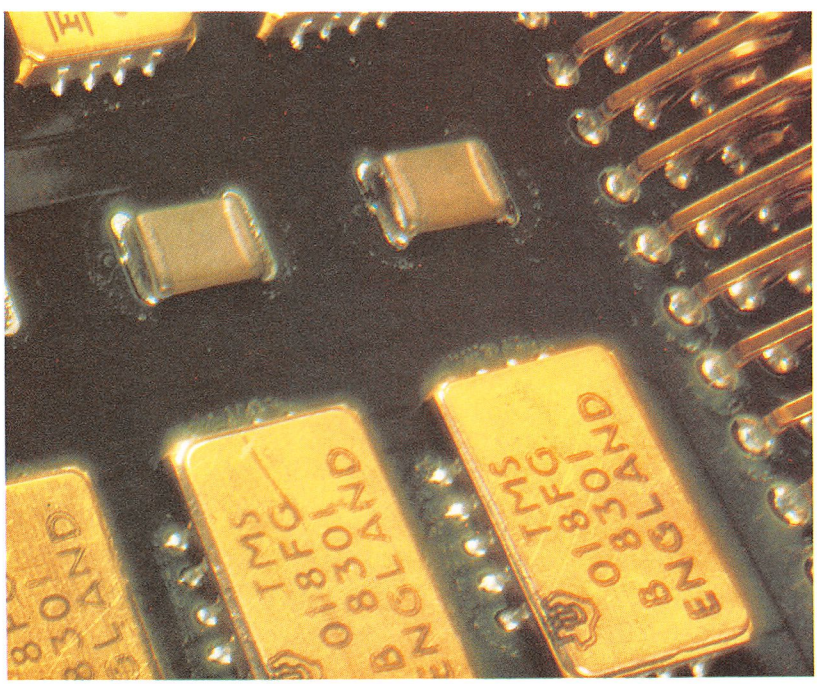

**3. Some microprocessors send out both a read/write and a memory enable signal** to control the transistor of data to and from memory.

**4. Microprocessors sometimes treat input/output devices as a system separate from memory.** In this case, an I/O enable and a R/W signal have to be used to control the I/O device.

**5. Interrupt signals** allow a microprocessor's normal operation to be temporarily halted and a special request attended to. Interrupt service routines, held in memory, contain the instructions to deal with each type of interrupt.




**Below:** part of a thick-film hybrid integrated circuit– one of the most modern methods of small volume circuit manufacture.



The Research House/Ericsson

receive or send information whenever desired.

## Power connections

As well as having signals and connections to control the microprocessor's operation, every electronic system needs to be connected to a power supply.

Many early microprocessors, made with MOS technology require three power supply voltages: + 12 V; – 12 V; + 5 V; as well as a ground connection, for example. More modern designs only need a nominal + 5 V voltage and a ground connection. These power connections are usually made through conductors formed as part of a printed circuit board (PCB). These conductors have to be of a low resistance and the connections have to be sound, otherwise errors due to electrical noise will occur.

## Program instructions

Having now ensured that all the microprocessor system's connections are made, we need to tell it what to do – by entering a sequence of instructions into the program memory. The instructions must of course be taken from the instruction set of the particular microprocessor used. Generating a microprocessor's program may not be as straightforward as connecting the system together, but with a little experience, most people can catch on quite quickly.

Any instruction given to a computer must be written in the digital code of the machine, so it can *sense* it, *decide* which it is, and *act*, to execute the instruction. The digital code that the machine understands is called **machine code**, and instructions can be written directly in this. If this is done then the program is said to be written in **machine language**, and is called a machine language program.

*Figure 6* shows where machine language takes its place in the hierarchy of programming languages – at the bottom. As you must know, machine language is the code that programs written in high level language like PASCAL, BASIC or FORTRAN are translated into for computers' use. This conversion is carried out by special software routines called **compilers**. However, these programs have to be written and stored inside computers.

A simple microprocessor based microcomputer system like the one we are discussing would not have a compiler, but the machine code still has to be moved one step closer to human language input. This is done by using **assembly language**, in which instructions are represented by mnemonic codes. Once written in mnemonics the program is fed into an **assembler** program, which makes the conversion to machine code and arranges the instructions in memory in the proper order.

### Microprocessor instruction set

The mnemonics that make up a microprocessor's assembly language 'vocabulary' are known as its **instruction set**. The instructions that a microprocessor executes are very basic operations – as you can see in *figure 7*. By using these simple instructions in the right combination, the microcomputer system can be made to operate in almost any way desired. *Figure 7* shows the different subsets that the instructions are broken down into: arithmetic; logical; comparison; branch; and data movement or transfer. Most of the instructions are self explanatory, but lets look at the data movement and arithmetic commands to gain an overall

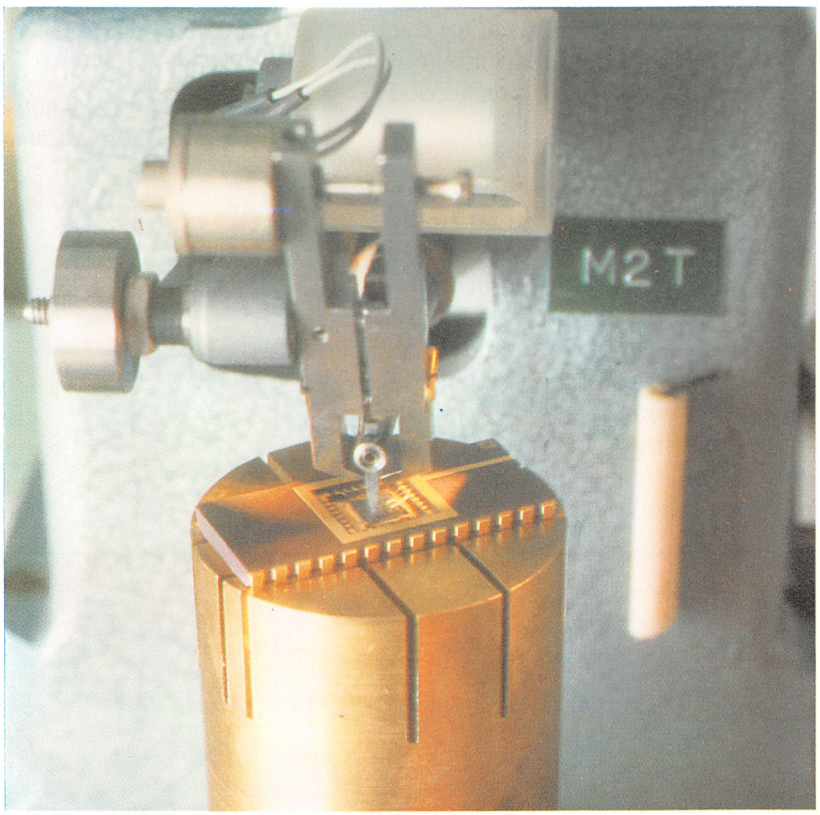
6

| Type of program | Program conversions required |
|---|---|
| **High-level language program**–Programmer writes program in a powerful general purpose language that is organised in a way directly related to the way humans solve problems. This is converted to the machine language program with a compiler. | High level language program → Compiler → Assembly language program → Assembler → Machine language program (Machine code) |
| **Assembly-language program**–Programmer writes the program using instruction abbreviations called mnemonics. This is converted to a machine language program with an assembler. | |
| **Machine-language program**–Programmer writes the program directly in terms of the binary codes that computer can understand (decode) and execute. No further conversion is necessary. | |

**6. The programming language hierarchy.**

7

| Arithmetic | Logical | Branch | Data movement or transfer | Comparison |
|---|---|---|---|---|
| Adds<br>Subtracts<br>Absolute value<br>Negation<br>Multiply<br>Divide<br>Shifts | AND<br>OR<br>NOT<br>Exclusive-Or | Unconditional<br>Conditional<br>Subroutine | Move<br>Load<br>Store | |




The Research House/Ericsson

**7. A microprocessor's instruction set** is made up of five basic types of command.

**Above:** automatic wire bonding in progress – a stage in the manufacture of a microprocessor package.

view of their operation.

### Data movement instructions

Data movement or transfer instructions move data from one memory location or register to another, within the microcomputer system. They can, for example:
1) move data from memory to a register inside the microprocessor to prepare for additional operations;
2) move data from a microprocessor register to memory or to an output;
3) move program constants from program memory to initialize microprocessor registers;
4) initialize data memory locations or output unit registers with constant values;
5) move data from inputs to a register inside the microprocessor.

Most microprocessors support all of these data movement operations. Some call these operations load or store, others simply refer to them as movement instructions.

Regardless of what the microprocessor manufacturers call these instructions, a short-hand is required to list them in a program sequence. This is where the mnemonics come in. The mnemonics are very short (at most three or four letters long) abbreviations for the various operations, such as: MOV for movement instructions; LD for the load operation; and ST for the store operation. Others are SWP for swap and XCHG for exchange. There is of course one mnemonic for each instruction.

### Arithmetic instructions and number codes

Microprocessors typically offer addition and subtraction as the basic arithmetic operations, but some offer multiplication, division, negation (sign change) and absolute value as well. Most modern microprocessors offer all the functions and also increment and decrement. The mnemonics for most of these instructions are rather obvious:
1) A or AD or ADD for addition;
2) S or SU or SB for subtraction;
3) MPY for multiply;
4) DIV for divide;
5) INC or INR for increment;
6) DEC or DCR for decrement;
7) NEG for change sign;
8) ABS for absolute value.

In arithmetic operations, numbers (data) are used as operands. For the microprocessor to be able to work with decimal numbers, they must be converted to binary code, binary coded decimal (BCD) or hexadecimal codes. To recap on the ways that computers work with numbers of different bases, refer back to *Basic Computer Science 3*.

### Operands

To make the instructions complete, more information must be included with the mnemonic to indicate which number, register or memory location contents are to

be 'operated on' by operation called for by the instruction. For example: MOV R1, R2 means move the contents of register 1 to register 2. R1 and R2 are the **operands** of the instruction. MOV is the data movement operation. Instructions like this can also be represented by symbolic diagrams to describe what the instructions mean. For example:

MOV R1, R2

is summarised by:

(R1) → (R2).

The parentheses mean 'the contents' and the arrow means 'move', so the instruction stands for 'move the contents of register 1 into register 2'. This parenthesis notation is important, because the microprocessor must be able to differentiate between the contents of a memory word and the address. The microprocessor goes to an address to locate the place of storage – what is contained in that place of storage is the contents. In the MOV R1, R2 instruction, register R1 may be a storage location (address) identified by the 16-bit code 0000 0101 0000 0001, but the contents of register R1 used for the move operation might be the decimal number 32, represented by the binary code 0010 0000.

8

| Assembly-language program | Machine-language program | Meaning of instruction |
|---|---|---|
| CMA | 0010 1111 | Complement A register |
| MOV B, A | 0100 0111 | Move contents of B Register to A Register |
| INR A | 0011 1100 | Increment contents of A Register |

**8. Part of an assembly language program**, showing the corresponding machine code and the instruction explanation.

### Assembling the instructions

*Figure 8* shows part of an assembly language program, the corresponding machine code and the literal meanings of each instruction. We now know that mnemonics can be converted to machine code by the assembler, so once the program is complete, all we have to do is input the instructions via a keyboard. The conversion to machine code, if desired, may be carried out manually.

## Glossary

| | |
|---|---|
| **assembler** | a software routine that prepares a machine language program, from a symbolic language (mnemonic) program |
| **decoder** | a combinational circuit that receives several parallel inputs, 'recognises' one or more patterns of input bits and puts out specific parallel bit patterns when these combinations are recognised |
| **enable** | to 'switch on' a chip or device |
| **interrupt signal** | signal that halts the operation of a process in order to carry out another. Upon completion of this task the original process will be returned to |
| **machine code** | operation code that a machine (microprocessor) is designed to recognise |
| **memory mapped I/O** | when a microprocessor doesn't have a separate input/output system, it assumes that the I/O devices are assigned to memory locations. Addressing these memory locations thus addresses the I/O devices |
| **mnemonics** | abbreviated instructions, in assembly language, standing for a microprocessor's operating commands |